UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA
DOUTORADO EM ENGENHARIA DE TELEINFORMÁTICA

CÉSAR LINCOLN CAVALCANTE MATTOS

RECURRENT GAUSSIAN PROCESSES AND ROBUST DYNAMICAL MODELING

FORTALEZA
2017

CÉSAR LINCOLN CAVALCANTE MATTOS

RECURRENT GAUSSIAN PROCESSES AND ROBUST DYNAMICAL MODELING

Tese apresentada ao Curso de Doutorado em Engenharia de Teleinformática do Programa de Pós-Graduação em Engenharia de Teleinformática do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do título de doutor em Engenharia de Teleinformática. Área de Concentração: Sinais e sistemas

Orientador: Prof. Dr. Guilherme de Alencar Barreto

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

M39r Mattos, César Lincoln.
Recurrent Gaussian Processes and Robust Dynamical Modeling / César Lincoln Mattos. – 2017.
189 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2017.
Orientação: Prof. Dr. Guilherme de Alencar Barreto.

1. Processos Gaussianos. 2. Modelagem dinâmica. 3. Identificação de sistemas não-lineares. 4. Aprendizagem robusta. 5. Aprendizagem estocástica. I. Título.

CDD 621.38

CÉSAR LINCOLN CAVALCANTE MATTOS

RECURRENT GAUSSIAN PROCESSES AND ROBUST DYNAMICAL MODELING

A thesis presented to the PhD course in Teleinformatics Engineering of the Graduate Program on Teleinformatics Engineering of the Center of Technology at Federal University of Ceará in fulfillment of the the requirement for the degree of Doctor of Philosophy in Teleinformatics Engineering.
Area of Concentration: Signals and systems

Approved on: August 25, 2017

EXAMINING COMMITTEE

Prof. Dr. Guilherme de Alencar Barreto
(Supervisor)
Universidade Federal do Ceará (UFC)

Prof. Dr. Antônio Marcos Nogueira Lima
Universidade Federal de Campina Grande
(UFCG)

Prof. Dr. Oswaldo Luiz do Valle Costa
Universidade de São Paulo (USP)

Prof. Dr. José Ailton Alencar Andrade
Universidade Federal do Ceará (UFC)

For my beloved family who always supports me: Fernando Lincoln, Carmen and Fernanda.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Guilherme de Alencar Barreto, for his long-time support, partnership and academic guidance. Guilherme knows a fine balance between a focused supervision and the freedom to explore and experiment. His patience, encouragement and dedication throughout all the PhD journey were invaluable.

I thank my many friends in the PPGETI, for all the intra and extra campus conversations about the most diverse topics. You were very important to proportionate a suitable environment for all this period. I specially thank José Daniel Alencar and Amauri Holanda, with whom I had many valuable discussions.

I am very grateful to Prof. Neil Lawrence, for kindly receiving me in his research group in the University of Sheffield. Neil has been a great intellectual inspiration and provided a significant boost to my studies. The way he works in his group and handles machine learning research will always have an influence on me.

I was very lucky to be around Neil's ML group in SITraN for six months. I learned so much from the incredible people I met there. All the experiences we shared were critical for my adaptation and will be treasured. Thanks a lot, everyone! A special thanks to my co-authors, Andreas Damianou and Zhenwen Dai, who dedicated a lot of time to help me with my research (and are great guys!).

I acknowledge the financial support of FUNCAP (for my PhD scholarship) and CAPES (for my time abroad via the PDSE program).

Finally, this thesis is dedicated to my family, Fernando Lincoln, Carmen and Fernanda, whose support was fundamental to the completion of this work.

"It is not knowledge, but the act of learning,
not possession but the act of getting there,
which grants the greatest enjoyment."

(Carl Friedrich Gauss)

RESUMO

O estudo de sistemas dinâmicos encontra-se disseminado em várias áreas do conhecimento. Dados sequenciais são gerados constantemente por diversos fenômenos, a maioria deles não passíveis de serem explicados por equações derivadas de leis físicas e estruturas conhecidas. Nesse contexto, esta tese tem como objetivo abordar a tarefa de identificação de sistemas não lineares, por meio da qual são obtidos modelos diretamente a partir de observações sequenciais. Mais especificamente, nós abordamos cenários desafiadores, tais como o aprendizado de relações temporais a partir de dados ruidosos, dados contendo valores discrepantes (*outliers*) e grandes conjuntos de dados. Na interface entre estatísticas, ciência da computação, análise de dados e engenharia encontra-se a comunidade de aprendizagem de máquina, que fornece ferramentas poderosas para encontrar padrões a partir de dados e fazer previsões. Nesse sentido, seguimos métodos baseados em Processos Gaussianos (PGs), uma abordagem probabilística prática para a aprendizagem de máquinas de *kernel*. A partir de avanços recentes em modelagem geral baseada em PGs, introduzimos novas contribuições para o exercício de modelagem dinâmica. Desse modo, propomos a nova família de modelos de Processos Gaussianos Recorrentes (RGPs, da sigla em inglês) e estendemos seu conceito para lidar com requisitos de robustez a *outliers* e aprendizagem estocástica escalável. A estrutura hierárquica e latente (não-observada) desses modelos impõe expressões não-analíticas, que são resolvidas com a derivação de novos algoritmos variacionais para realizar inferência determinista aproximada como um problema de otimização. As soluções apresentadas permitem a propagação da incerteza tanto no treinamento quanto no teste, com foco em realizar simulação livre. Nós avaliamos em detalhe os métodos propostos com *benchmarks* artificiais e reais da área de identificação de sistemas, assim como outras tarefas envolvendo dados dinâmicos. Os resultados obtidos indicam que nossas proposições são competitivas quando comparadas a modelos disponíveis na literatura, mesmo nos cenários que apresentam as complicações mencionadas, e que a modelagem dinâmica baseada em PGs é uma área de pesquisa promissora.

Palavras-chave: Processos Gaussianos. Modelagem dinâmica. Identificação de sistemas não-lineares. Aprendizagem robusta. Aprendizagem estocástica.

ABSTRACT

The study of dynamical systems is widespread across several areas of knowledge. Sequential data is generated constantly by different phenomena, most of them that we cannot explain by equations derived from known physical laws and structures. In such context, this thesis aims to tackle the task of nonlinear system identification, which builds models directly from sequential measurements. More specifically, we approach challenging scenarios, such as learning temporal relations from noisy data, data containing discrepant values (*outliers*) and large datasets. In the interface between statistics, computer science, data analysis and engineering lies the machine learning community, which brings powerful tools to find patterns from data and make predictions. In that sense, we follow methods based on Gaussian Processes (GP), a principled, practical, probabilistic approach to learning in *kernel* machines. We aim to exploit recent advances in general GP modeling to bring new contributions to the dynamical modeling exercise. Thus, we propose the novel family of Recurrent Gaussian Processes (RGPs) models and extend their concept to handle outlier-robust requirements and scalable stochastic learning. The hierarchical latent (non-observed) structure of those models impose intractabilities in the form of non-analytical expressions, which are handled with the derivation of new variational algorithms to perform approximate deterministic inference as an optimization problem. The presented solutions enable uncertainty propagation on both training and testing, with focus on free simulation. We comprehensively evaluate the introduced methods with both artificial and real system identification benchmarks, as well as other related dynamical settings. The obtained results indicate that our propositions are competitive when compared to models available in the literature within the aforementioned complicated setups and that GP-based dynamical modeling is a promising area of research.

Keywords: Gaussian Processes. Dynamical modeling. Nonlinear system identification. Robust learning. Stochastic learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS AND ACRONYMS

ARD	Automatic Relevance Determination
BFGS	Broyden-Fletcher-Goldfarb-Shanno (algorithm)
EM	Expectation Maximization
EP	Expectation Propagation
ESGP	Echo State GP
ESN	Echo State Network
GP	Gaussian Process
GP-LEP	GP model with Laplace likelihood and EP inference
GP-NARX	GP model with NARX structure
GP-RLARX	Robust GP Latent Autoregressive Model
GP-SSM	GP model with SSM structure
GP-tVB	GP model with Student-t likelihood and variational inference
GPDM	Gaussian Process Dynamical Model
GP-LVM	Gaussian Process Latent Variable Model
KL	Kullback-Leibler (divergence)
KLMS	Kernel Least Mean Squares
KRLS	Kernel Recursive Least-Squares
LSTM	Long Short-Term Memory
MAP	Maximum a Posteriori
MCMC	Markov Chain Monte Carlo
MIMO	Multiple-Input and Multiple-Output
ML	Maximum Likelihood
MLP	Multilayer Perceptron
MLP-NARX	MLP with NARX structure
NAR	Nonlinear Autoregressive
NARMAX	Nonlinear Autoregressive Moving Average with eXogenous inputs
NARX	Nonlinear Autoregressive with eXogenous inputs
NFIR	Nonlinear Finite Impulse Response
NLPD	Negative Log-Predictive Density
NN	Neural Network
NOE	Nonlinear Output Error

OE	Output Error
PAM	Partition Around Medoids
PCA	Principal Component Analysis
RBF	Radial Basis Function
ReLu	Rectifier Linear Unit
REVARB	REcurrent VARiational Bayes
REVARB-t	REVARB with Student-t likelihood
RGP	Recurrent Gaussian Process
RGP-t	RGP with Student-t likelihood
RKHS	Reproducing Kernel Hilbert Spaces
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SISO	Single-Input and Single-Output
SISOG	System Identification using Sparse Online GP
SMC	Sequential Monte Carlo
SNR	Signal to Noise Ratio
S-REVARB	Stochastic REVARB
SSM	State-Space Model
SVI	Stochastic Variational Inference
SVI-GP	SVI for sparse GP models
SVM	Support Vector Machines
TBPTT	Truncated Backpropagation Through Time
VB	Variational Bayes
VGPDS	Variational Gaussian Process Dynamical Systems

LIST OF SYMBOLS

$\langle\rangle_{p(\cdot)}$	Expectation with respect to the distribution $p(\cdot)$
$\boldsymbol{D}$	Input dimension
$\mathbb{E}\{\cdot\}$	Mean of the random variable in the argument
$\boldsymbol{\varepsilon}_i^{(h)}$	Noise variable related to the i-th instant of the h-th layer
$\boldsymbol{\zeta}_m^{(h)}$	j-th pseudo-input of the h-th layer
$f(\cdot)$	Unknown (possibly) nonlinear function
f_i	Latent function value in the i-th instant
f_*	Latent function value related to a test input
$\boldsymbol{H}$	Number of hidden layers
$\boldsymbol{I}$	Identity matrix
$k(\cdot,\cdot)$	Covariance (or kernel) function
$\boldsymbol{K}_{fz}$	Covariance (or kernel) matrix between the vectors $\boldsymbol{f}$ and $\boldsymbol{z}$
$\boldsymbol{K}_f$	Covariance (or kernel) matrix related to the vector $\boldsymbol{f}$
$\mathrm{KL}(\cdot\|\|\cdot)$	Kullback-Leibler divergence
$\boldsymbol{L}$	Order of the lagged latent dynamical variables
$\boldsymbol{L}_u$	Order of the lagged exogenous inputs
$\boldsymbol{L}_y$	Order of the lagged outputs
$\boldsymbol{\lambda}_i^{(h)}$	Variational variance related to the i-th dynamical latent variable of the h-th layer
$\log$	Natural logarithm operator
$\boldsymbol{M}$	Number of pseudo-inputs
$\boldsymbol{\mu}_i^{(h)}$	Variational mean related to the i-th dynamical latent variable of the h-th layer
$\boldsymbol{N}$	Number of estimation (or training) samples
$p(\cdot)$	Probability density function
$q(\cdot)$	Variational distribution
$\boldsymbol{Q}$	Product of all the modeled variational distributions

σ_h^2	Variance of the noise related to the h-th layer
Tr	Trace operator
$\boldsymbol{\theta}$	Vector of parameters or hyperparameters
τ_i	Precision (inverse variance) related to the i-th observation
u_i	Exogenous system input in the i-th instant
$\mathbb{V}\{\cdot\}$	Variance of the random variable in the argument
w_d	d-th inverse lengthscale hyperparameter
$x_i^{(h)}$	Dynamical latent variable in the i-th instant of the h-th layer
$\bar{\boldsymbol{x}}_i^{(h)}$	Latent regressor vector in the i-th instant of the h-th layer
$\hat{\boldsymbol{x}}_i^{(h)}$	Input vector in the i-th instant of the h-th layer
$\boldsymbol{x}_*$	Test input vector
$\hat{\boldsymbol{X}}^{(h)}$	Stack of all the vectors $\hat{\boldsymbol{x}}_i^{(h)}$
y_i	System observed output in the i-th instant
y_*	System observed output related to a test output
$\boldsymbol{z}^{(h)}$	Vector of inducing points

CONTENTS

1 INTRODUCTION

"The most important questions of life are indeed,
for the most part, only problems of probability."
(Pierre-Simon Laplace)

The contemporary world is immersed in a great variety of systems and subsystems with intricate interactions between themselves and with our society. Some of those systems are related to fixed input-output mappings and, therefore, only the current interactions affect their operation. Those are labeled as *static* systems. However, the class of *dynamical* systems do not present those properties and require distinct study strategies.

The concept of a dynamical system defines a process whose state presents a temporal dependence. In other words, the output of the system depends not only on the current external inputs but also their previous values. In many cases the external stimuli is easy to define, for instance, the opening of a valve in a hydraulic actuator or the wind speed in a wind turbine. Alternatively, if the external signals that excite the system are not observed, its measured outputs are usually called a *times series* (LJUNG, 1999). Those definitions may be used to describe phenomena of several different fields, such as engineering, physics, chemistry, biology, finance and sociology (CHUESHOV, 2002). Moreover, it is important to note that, ultimately, all real systems are dynamical, even though for some of them the static analysis is sufficient (AGUIRRE, 2007).

The analysis of a dynamical system demands that its behavior must be explained by a mathematical model, which can be obtained theoretically or empirically. The theoretical methodology is based on findings supported by equations relating parameters from a known structure, determined by underlying physical laws. On the other hand, the experimental approach, named *identification*, seeks an appropriate model from measurements and *a priori* knowledge, possibly acquired from past experiments (ISERMANN; MÜNCHHOF, 2011). The latter, which avoids the limitations and problem-specific complexities of the purely theoretical approach, is the main application subject of this work.

System identification, while of fundamental importance for the design, control and analysis of industrial processes in general, is a challenging task. Although both linear and nonlinear systems have been extensively studied for several decades, the latter

usually involve much more complex analyses, provided the wide class of possible model nonlinearities (BILLINGS, 2013). Since the quality of the model is frequently the bottleneck of the final problem solution (NELLES, 2013), complications such as noisy data and the presence of discrepant observations in the form of *outliers* must be carefully considered. Moreover, modern systems have generated increasingly larger datasets, a feature which by itself provides algorithmic and computational demands.

As described so far, the system identification exercise seems a perfect candidate to be tackled by the *machine learning* community, a field closely related to statistics and at the intersection of computer science, data analysis and engineering. The goal of machine learning is to apply methods that can detect patterns in data, make predictions and aid decision making (MURPHY, 2012). As opposed to other common machine learning applications, such as standard continuous regression (static mapping) and classification (mapping to discrete classes), we are interested in working with *sequential* records. Thus, all the models presented in this work can be viewed as machine learning efforts tailored to handle dynamical data. It is worth mentioning the recent *preprint* work by Pillonetto (2016), who analyzes general connections between the fields of system identification and machine learning, with focus on the shared problem of *learning from examples*.

In this thesis we mainly follow the *Bayesian* point of view, where the model accounts for the *uncertainty* in the noisy data and in the learned dynamics (PETERKA, 1981). More specifically, we apply Gaussian Processes (GP) models, a principled, practical, probabilistic approach to learning in *kernel* machines (RASMUSSEN; WILLIAMS, 2006). Differently from modeling approaches that aim to fit a parametrized structure to a system's inputs and outputs, a GP model directly describes the probabilistic relations between the output data with respect to the inputs (KOCIJAN, 2016). Thus, GP models have been widely applied to the so called "black box" approach[1] to system modeling (AŽMAN; KOCIJAN, 2011). The probabilistic information obtained *a posteriori* from GP models, i.e., after training using observed data, has made them an attractive stochastic tool to applications in control and system identification in general (KOCIJAN *et al.*, 2005).

The Bayesian nature of GP-based models considers the uncertainty inherited from the quality of the available data and the modeling assumptions, which enables a clear probabilistic interpretation of its predictions. Thus, an immediate advantage of the GP

[1] A black box is a system whose internal structure is unknown but its analysis is done directly from the relations of its inputs and measured outputs.

framework over other regression methods, like Neural Networks (NNs) or Support Vector Machines (SVMs), is that its outputs are distributions, i.e., instead of point estimates, each prediction is given by a fully defined probability distribution. Therefore, it explicitly indicates its uncertainty due to limited information about the process that generated the observed data and enables error bars in the predictions, a valuable feature in many applications, such as control and optimization. Gregorčič and Lightbody (2008) present some more advantages of GPs over other learning methods, such as the reduced number of hyperparameters, less susceptibility to the so called "curse of dimensionality"[2] and the more transparent analysis of the obtained results due to the uncertainty they are able to express.

Taking advantage of the aforementioned features, the authors have explored diverse applications of GP-based dynamical models, such as nonlinear signal processing (PÉREZ-CRUZ *et al.*, 2013), human motion modeling (EK *et al.*, 2007; WANG *et al.*, 2008; JIANG; SAXENA, 2014), speech representation and synthesis (HENTER *et al.*, 2012; KORIYAMA *et al.*, 2014), fault detection (JURICIC; KOCIJAN, 2006; OSBORNE *et al.*, 2012) and model-based control (KOCIJAN *et al.*, 2004; LIKAR; KOCIJAN, 2007; KO *et al.*, 2007; DEISENROTH; RASMUSSEN, 2011; ROCHA *et al.*, 2016; KLENSKE *et al.*, 2016; VINOGRADSKA *et al.*, 2016).

There is also a vast body of theses with diverse contributions to the use of GP-based methods to dynamical modeling and more in-depth theoretical and empirical analyses, such as GP with noisy inputs and derivative observations (GIRARD, 2004), multiple GPs and state-space time series models (NEO, 2008), implementation and application to engineering systems (THOMPSON, 2009), nonlinear filtering and change-point detection (TURNER, 2012). Recently, work in this topic has been even more active, with several authors proposing more flexible model structures, powerful learning algorithms and tackling task-specific issues in their doctoral researches (MCHUTCHON, 2014; FRIGOLA-ALCADE, 2015; DAMIANOU, 2015; SVENSSON, 2016; BIJL, 2016).

The present PhD work builds upon such rich literature on dynamical GP-based models in order to revisit the problem of nonlinear system identification. We aim to exploit recent advances in general GP modeling to bring new contributions to the field, as detailed in our objectives, listed as follows.

[2]Curse of dimensionality is a term usually applied to refer to the many difficulties that arise in the analysis of problems defined in high-dimensional spaces.

1.1 Objectives

The main objective of this thesis consists in elaborating and evaluating GP-based formulations capable of modeling dynamical systems in challenging scenarios, such as when facing noisy datasets, data containing outliers or very large training sets.

The specific objectives of our work are listed below:

1. To propose a novel GP-based model specifically designed to learn dynamics from sequential data, with the focus on accounting for data and model uncertainties over both training and testing;
2. To propose novel GP-based models to tackle the problem of learning dynamics from data containing non-Gaussian noise in the form of outliers;
3. To propose a novel inference methodology to enable stochastic training of dynamical GP models with large datasets;
4. To evaluate all the proposed models in their specific tasks by performing *free simulation* on unseen data, as described in the next section.

1.2 System Identification

The system identification task, i.e., the process of obtaining a model only from a system's inputs and outputs, can be summarized by the following major steps (LJUNG, 1999):

1. **Collect data**: Via specifically designed experiments, data is recorded from a real system, where the user chooses the measured signals, sampling rates and the inputs to excite its dynamics, aiming for the most informative records possible. Note however that sometimes one has to build a model only from the data obtained during the system normal operation.
2. **Determine model structure**: A sufficiently general class of models is chosen in order to constrain the search for a good model to explain the analyzed phenomenon. Such structure may contain adjustable parameters to enable variations in its behavior.
3. **Perform model selection**: The identification step, where the collected data is used to estimate unknown parameters, usually by assessing how well the model is able to reproduce the collected data, as quantified by a given rule.
4. **Validate the model**: The evaluation step consists in verifying if predictions made

with the chosen model are acceptable, preferably using data similar but not equal to the records used in the model selection phase, in order to evaluate the model *generalization* capability. The validation metric should be chosen accordingly the final model application.

As many engineering procedures, the previous steps may be repeated until one is satisfied with the obtained results, revising each step after analyzing encountered issues. In the present thesis we are mostly interested in the latter three steps, since the data collection step is considered to have been already done. We refer the readers to Ljung (1999), Zhu (2001), Aguirre (2007), Isermann and Münchhof (2011) to learn more about such data recording step, also called *experiment design*.

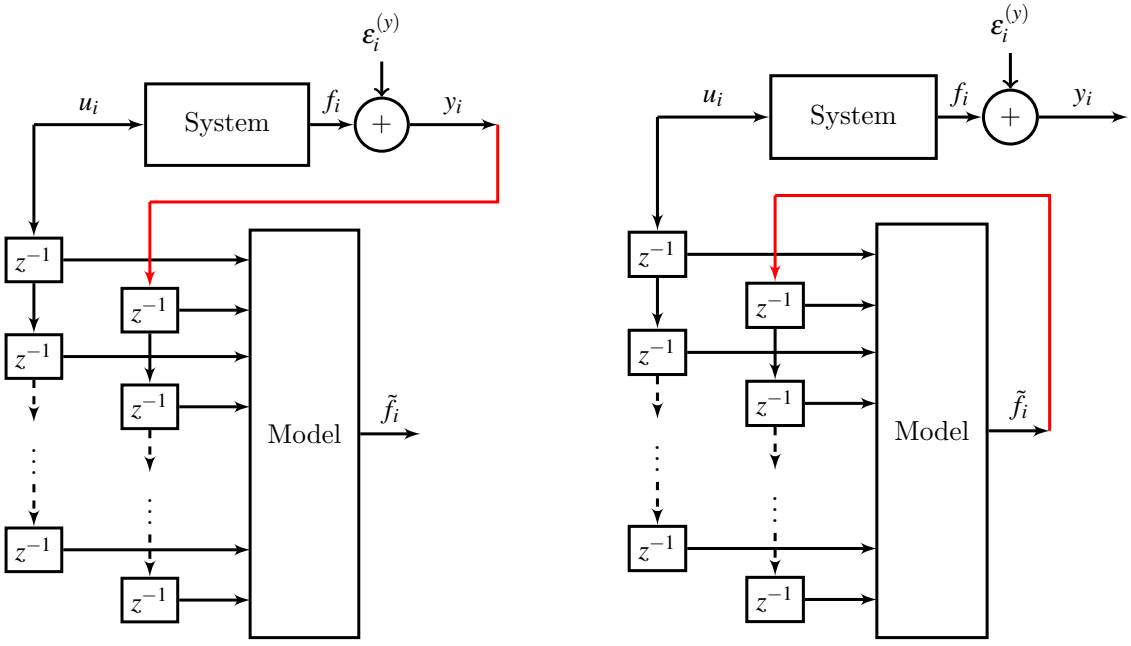
Regarding Step 4 of the aforementioned system identification methodology, i.e., where the selected dynamical model is used to perform predictions using separate test data, two common evaluation approaches are described below (NELLES, 2013):

One-step-ahead prediction The next prediction is based on previous test inputs and test observed outputs until the current instant, in a *feedforward* fashion. Typical applications include weather and stock market forecasting, where measures are readily available and applied especially for short-term prediction.

Free simulation Predictions are made only from past immediate test inputs and predictions, without using past test observations, following a *recurrent* structure. It is also called *infinite step ahead prediction*, *model predicted output* or simply *simulation*. Such procedure is necessary when the model will be used to replace the original system, for instance, to experiment with different inputs or as a way to diagnose a real process.

Fig. 1 illustrates the difference between the evaluation strategies described above, considering an autoregressive set-up. The external input of the i-th iteration is denoted as u_i, while the noiseless system output, noisy measurement (corrupted by the disturbance $\varepsilon_i^{(y)}$) and model prediction are respectively f_i, y_i and $\tilde{f}_i$. The z^{-1} blocks indicate unit delays and the red lines highlight the difference between the two diagrams.

Although one-step-ahead prediction can be useful for some tasks, it is argued by Billings (2013) that such validation methodology can be misleading, because *even poor models can look good*. Thus, throughout this thesis we follow the more challenging free simulation approach.



(a) One-step ahead prediction. (b) Free simulation.

Figure 1 – Block diagrams of common evaluation methodologies for system identification. The z^{-1} blocks indicate unit delays. The red lines highlight the difference between both approaches. Note that the left diagram presents a feedforward configuration, as opposed to the diagram in the right, which is recurrent.

1.3 Probabilistic Reasoning

We mostly adopt in this thesis a probabilistic approach to modeling[3]. Its main ingredient is the account for uncertainty, which comes from noisy observations, finite data and our limited understanding about the analyzed phenomenon (BISHOP, 2006; BARBER, 2012). Those characteristics are intrinsic to machine learning in general, which turns probability an important tool for the area. Although in the present thesis we are not directly interested in discussing about the basic meanings and interpretations about the very much vague concept of probability, it is worth mentioning some of the predominant views.

One possible view is to consider probability as being the *long run* frequency of events, i.e., the probability of a certain event would be given by its frequency in the limit of a very large number of trials (MURPHY, 2012). That is called the *frequentist* interpretation or simply *frequentism*.

[3]Strictly speaking, we follow a *statistical modeling* approach, since we have an unknown model and some observations, while the *probabilistic perspective* would deal only with random variables and their relations (JORDAN, 2003). However, the machine learning literature commonly applies the probabilistic modeling label to refer to models that handle (at least some of) its components with probability distributions (BISHOP, 2006; MURPHY, 2012). Thus, the later will also be the term used throughout this thesis.

Another view treats probability as the quantification of our *belief* about something. In this approach, the *Bayesian* view, the uncertainty of an event is related to the degree of information we have about it (JEFFREYS, 1998; JAYNES, 2003; COX, 2006). As already mentioned, this is the approach followed by the present work.

The fundamental step in Bayesian modeling is to perform *reasoning*, or *inference*, i.e., to learn a probability model given a dataset and output probability distributions on unobserved quantities, such as components of the model itself or predictions for new observations (GELMAN *et al.*, 2014a).

From a methodological point of view, it is interesting to separate the concepts of *model* and *learning algorithm*. While the first is the mathematical way of representing knowledge, for instance, in a probabilistic way, the latter is the procedure in which we perform inference with our model. Such separation is important to enable model and learning algorithm analyses and improvements independently of each other, as well as to apply different inference methods to the same model in distinct applications (KOLLER; FRIEDMAN, 2009).

As we have already declared, in this thesis our main interest lies in the GP modeling approach, which is first of all a *Bayesian nonparametric* framework. The first term of such label indicates, as previously discussed, that it treats the uncertainty of the data and the model itself by probability means. The second term indicates the absence of a finite set of *parameters*, i.e., the knowledge captured by the model is not concentrated in a fixed set of tunable weights, such as in neural networks or models with weighted basis functions[4]. Furthermore, the complexity of a GP model grows with the amount of data made available. In other words, instead of a rigid prespecified structure, the model allows for the data to "speak by itself".

1.4 A Word About Notation and Nomenclature

We have tried to maintain a certain degree of formalism in the mathematical descriptions of the current work, while constantly reminding ourselves of our practical goals over pure analytical derivations. Nevertheless, we could not avoid the common problem of choosing and keeping a comprehensive mathematical notation. In order to avoid confusion,

[4] Although there are alternative (and conflicting) definitions of parametric and nonparametric models in the literature, throughout this work we consider the interpretation presented here, more common in the machine learning community (BISHOP, 2006; RASMUSSEN; WILLIAMS, 2006; MURPHY, 2012).

we refer the readers to the List of Symbols at the beginning of this document, which is complemented by the additional conventions below, adopted throughout this thesis.

- We follow a notation for random variables common in the the probabilistic modeling literature, used for instance by Neal (1994), Särkkä (2013), Gelman *et al.* (2014a). Thus, a vector $\boldsymbol{f}$ with a multivariate Gaussian distribution defined by its mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{K}$ is denoted as $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$ or equivalently via its explicit density function $p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{K})$. Note that, in the latter, the symbol '|' is used to highlight the random variable associated with the distribution and separate it from other quantities.
- Although probability density functions are written using $p(\cdot)$, we use the notation $q(\cdot)$ to denote a *variational* distribution, i.e., a parametrized simpler function used to approximate a more complex density. A factorized variational distribution of multiple variables, e.g., $\boldsymbol{f}$ and $\boldsymbol{x}$, is sometimes compactly written as $Q = q(\boldsymbol{f})q(\boldsymbol{x})$.
- In probabilistic models we frequently have to marginalize random variables in joint or conditional distributions. Such operation consists in integrating out a variable from a given distribution considering all the possible values it can assume. For instance, to marginalize the variable $\boldsymbol{f}$ from the joint distribution $p(\boldsymbol{y}, \boldsymbol{f})$, we have to solve the integral $p(\boldsymbol{y}) = \int_{-\infty}^{\infty} p(\boldsymbol{y}, \boldsymbol{f}) \mathrm{d}\boldsymbol{f}$.

  In this thesis we rewrite the former integral in an equivalent but more compact way: $\int_{\boldsymbol{f}} p(\boldsymbol{y}, \boldsymbol{f})$, where the subindex in the integral symbol indicates the integrated variables over their respective domains, with the operation considering all the possible values it can take. This approach, adopted for instance by Barber (2012), will allow us to write long expressions avoiding notation clutter.

As important (and complicated) as maintaining a coherent notation is the issue of choosing a proper nomenclature. Since the machine learning area has always been at the interface of other disciplines, the terminology should be as clear as possible for readers from neighbor communities. For instance, we anticipate that the words *training*, *estimation* and *learning* will be used as synonyms. Also, *testing* will be equivalent to *evaluating*. Thus, we try our best to clearly explain all the possibly dubious terms along the next chapters.

1.5 List of Publications

The following works were developed during the course of the PhD period. The works marked with a "*" are the ones more directly related to the main subjects of the present thesis, but all of them were relevant to the overall academic formation.

1. **César L. C. Mattos**; José D. A. Santos & Guilherme A. Barreto (2014). *Classificação de padrões robusta com redes Adaline modificadas*, XI Encontro Nacional de Inteligência Artificial e Computacional, ENIAC 2014.
2. José D. A. Santos; **César L. C. Mattos** & Guilherme A. Barreto (2014). *A Novel Recursive Kernel-Based Algorithm for Robust Pattern Classification*, 15th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2014.
3. **César L. C. Mattos**; José D. A. Santos & Guilherme A. Barreto (2014). *Improved Adaline Networks for Robust Pattern Classification*, 24th International Conference on Artificial Neural Networks, ICANN 2014.
4. ***César L. C. Mattos**; José D. A. Santos & Guilherme A. Barreto (2015). *Uma avaliação empírica de modelos de processos Gaussianos para identificação robusta de sistemas dinâmicos*, XII Simpósio Nacional de Automação Inteligente, SBAI 2015.
5. José D. A. Santos; **César L. C. Mattos** & Guilherme A. Barreto (2015). *Performance Evaluation of Least Squares SVR in Robust Dynamical System Identification*, Lecture Notes in Computer Science, vol. 9095, pages 422-435.
6. ***César L. C. Mattos**; José D. A. Santos & Guilherme A. Barreto (2015). *An Empirical Evaluation of Robust Gaussian Process Models for System Identification*, Lecture Notes in Computer Science, vol. 9375, pages 172-180.
7. ***César L. C. Mattos**; Andreas Damianou; Guilherme A. Barreto & Neil D. Lawrence (2016). *Latent Autoregressive Gaussian Processes Models for Robust System Identification*, 11th IFAC Symposium on Dynamics and Control of Process Systems DYCOPS-CAB 2016.
8. ***César L. C. Mattos**; Zhenwen Dai; Andreas Damianou; Jeremy Forth; Guilherme A. Barreto & Neil D. Lawrence (2016). *Recurrent Gaussian Processes*, International Conference on Learning Representations, ICLR 2016.
9. **César L. C. Mattos**; Guilherme A. Barreto & Gonzalo Acuña (2017). *Randomized Neural Networks for Recursive System Identification in the Presence of Outliers: A*

Performance Comparison, 14th International Work-Conference on Artificial Neural Networks, IWANN 2017.

10. **César L. C. Mattos**; Guilherme A. Barreto; Dennis Horstkemper & Bernd Hellingrath (2017). *Metaheuristic Optimization for Automatic Clustering of Customer-Oriented Supply Chain Data*, WSOM+ 2017.
11. ***César L. C. Mattos**; Zhenwen Dai; Andreas Damianou; Guilherme A. Barreto & Neil D. Lawrence (2017). *Deep Recurrent Gaussian Processes for Outlier-Robust System Identification*, Journal of Process Control.
12. ***César L. C. Mattos** & Guilherme A. Barreto (2017). *A Stochastic Variational Framework for Recurrent Gaussian Processes Models*, In submission.

1.6 Organization of the Thesis

The remaining of this document is organized as follows:

- Chapter 2 introduces the Bayesian GP modeling framework, first detailing its application to standard regression and then describing important variations and enhancements that will be used in our work, such as sparse approximations, GP models with uncertain inputs and deep GPs;
- Chapter 3 reviews GP-based contributions for dynamical modeling in the literature and details the proposed Recurrent Gaussian Processes (RGP) model. It also contains the presentation of the variational framework called Recurrent Variational Bayes (REVARB), used to perform inference with RGPs. Afterwards, comprehensive computational experiments are presented to evaluate the newly introduced approach.
- Chapter 4 tackles the problem of learning dynamics from data containing non-Gaussian noise in the form of outliers. GP variants for robust regression found in the literature are evaluated in this task and two novel robust GP-based models are proposed. Modified variational procedures are derived for those new approaches, which are then evaluated in several benchmarks.
- Chapter 5 approaches the challenge of learning dynamics with GP models from large sequential datasets, a known limitation of standard formulations. A novel stochastic recurrent inference framework is introduced for the RGP model and the two algorithms that implement it are evaluated in large scale system identification tasks.

- Chapter 6 concludes the thesis with a final discussion about our work and recommendations for further research.

We also present additional mathematical derivations in the Appendix.

2 THE GAUSSIAN PROCESS MODELING FRAMEWORK

"How dare we speak of the laws of chance?
Is not chance the antithesis of all law?"
(Joseph Bertrand)

GP models have been used for predictions by the geostatistics community for many decades, where is usually known as *kriging* (MATHERON, 1973; STEIN, 2012). It was only some years later that works such as Williams and Rasmussen (1996) and Rasmussen (1996) indicated that GP models are capable to outperform more conventional nonlinear regression methods, such as neural networks (NN) (Bayesian NNs and ensembles of NNs) and spline methods. Since then, the GP framework has also been applied to classification problems (WILLIAMS; BARBER, 1998; OPPER; WINTHER, 2000; KUSS; RASMUSSEN, 2005), unsupervised learning (LAWRENCE, 2004; TITSIAS; LAWRENCE, 2010) and Bayesian global optimization (SNOEK *et al.*, 2012). Furthermore, in Chapter 3 we will approach dynamical modeling with GPs and in Chapter 4 we will cover the use of GP models in the problem of learning from data containing outliers. All those different applications highlight the flexibility of the GP modeling approach.

In this chapter we describe the general GP framework with focus on its application to the standard regression task with noisy Gaussian distributed observations. We start by following the presentation made by Rasmussen and Williams (2006) and introduce the more convenient GP formulation on *function space*, but also detail the *parameter space* view, which explicits the link between GPs and other kernel-based methods. We then describe important extensions to standard GPs, such as sparse GPs, unsupervised GP learning and hierarchical modeling, some of the tools largely used throughout this work.

2.1 Multivariate Gaussian Distribution: Two Important Properties

Since most of the GP features are inherited from the multivariate Gaussian distribution, we shall first briefly review it. The notation was intentionally chosen to allow easy relation later to the GP learning setting.

Let a vector of N random variables $\boldsymbol{f} \in \mathbb{R}^N$ follow a multivariate Gaussian

distribution expressed by

$$p(\boldsymbol{f}|\boldsymbol{\mu},\boldsymbol{K}_f) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu},\boldsymbol{K}_f) = \frac{1}{(2\pi)^{\frac{N}{2}}|\boldsymbol{K}_f|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{f}-\boldsymbol{\mu})^\top \boldsymbol{K}_f^{-1}(\boldsymbol{f}-\boldsymbol{\mu})\right), \tag{2.1}$$

where $|\cdot|$ denotes the determinant of a matrix and the distribution is completely defined by its mean vector $\boldsymbol{\mu} \in \mathbb{R}^N$ and its covariance matrix $\boldsymbol{K}_f \in \mathbb{R}^{N\times N}$.

Consider now $\boldsymbol{f}_1 \in \mathbb{R}^{N_1}$ and $\boldsymbol{f}_2 \in \mathbb{R}^{N_2}$, $N = N_1 + N_2$, two subsets of the vector $\boldsymbol{f}$, which are jointly Gaussian:

$$\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{K}_f), \quad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}, \quad \boldsymbol{K}_f = \begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{12}^\top & \boldsymbol{K}_{22} \end{bmatrix},$$

where the vectors and matrices have the appropriate dimensions. We shall emphasize two fundamental properties of such collection of random variables:

Marginalization The observation of a larger collection of variables does not affect the marginal distribution of smaller subsets, which, given the former expressions, implies that $\boldsymbol{f}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{K}_{11})$ and $\boldsymbol{f}_2 \sim \mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{K}_{22})$.

Conditioning Conditioning on Gaussians results in a new Gaussian distribution given by

$$p(\boldsymbol{f}_1|\boldsymbol{f}_2) = \mathcal{N}(\boldsymbol{f}_1|\boldsymbol{\mu}_1 + \boldsymbol{K}_{12}\boldsymbol{K}_{22}^{-1}(\boldsymbol{f}_2-\boldsymbol{\mu}_2), \boldsymbol{K}_{11} - \boldsymbol{K}_{12}\boldsymbol{K}_{22}^{-1}\boldsymbol{K}_{12}^\top). \tag{2.2}$$

Although simple, both properties are of fundamental importance to formulate most of the GP modeling framework in the next sections.

2.2 GP Prior over Functions

A GP is formally defined as a distribution over functions $f : \mathscr{X} \to \mathbb{R}$ such as that, for any finite subset $\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\} \subset \mathscr{X}$, $\boldsymbol{x}_i \in \mathbb{R}^D$, of the domain $\mathscr{X}$, a vector $\boldsymbol{f} = [f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_N)]^\top$ follows a multivariate Gaussian distribution $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu},\boldsymbol{K}_f)$. By viewing functions as infinitely long vectors, all points generated by $f(\cdot)$ are jointly modeled as a single GP, an infinite object.

Fortunately, because of the marginalization propriety of Gaussians, we can analyze such object by working with a finite multivariate Gaussian distribution over the vector $\boldsymbol{f} \in \mathbb{R}^N$. Considering N samples of D-dimensional inputs $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, i.e., a stack of the vectors $\boldsymbol{x}_i|_{i=1}^N$, we have

$$\boldsymbol{f} = f(\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{0},\boldsymbol{K}_f), \tag{2.3}$$

where we have defined a zero mean vector $\mathbf{0} \in \mathbb{R}^N$ for the GP prior. Any other value, or even another model, could be chosen for the mean, but our choice is general enough and will be used all along this thesis. We shall see as follows that a zero mean prior does not correspond to zero mean posterior.

In order to model the function values $\boldsymbol{f}$ with respect to different inputs, the elements of the covariance matrix $\boldsymbol{K}_f$ are calculated by $[K_f]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j), \forall i, j \in \{1, \cdots, N\}$, where $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^D$ and $k(\cdot, \cdot)$ is the so-called covariance (or *kernel*) function, which is restricted so that it generates a positive semidefinite matrix $\boldsymbol{K}_f$, also called the *kernel* matrix. The *exponentiated quadratic* kernel (also named *squared exponential* or Radial Basis Function (RBF)), which enforces a certain degree of smoothness, is a common choice and will be applied throughout this work:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d^2 (x_{id} - x_{jd})^2\right], \tag{2.4}$$

where x_{id}, x_{jd} are respectively the d-th components of the inputs $\boldsymbol{x}_i, \boldsymbol{x}_j$ and $\boldsymbol{\theta} = [\sigma_f^2, w_1^2, \ldots, w_D^2]^\top$ is the collection of kernel *hyperparameters*, responsible for characterizing the covariance of the model. For instance, the values of $w_1^2, \ldots, w_D^2$, the *inverse lengthscales*, are responsible for the so-called *automatic relevance determination* (ARD) of the input dimensions, where the hyperparameters related to less relevant dimensions have low values.

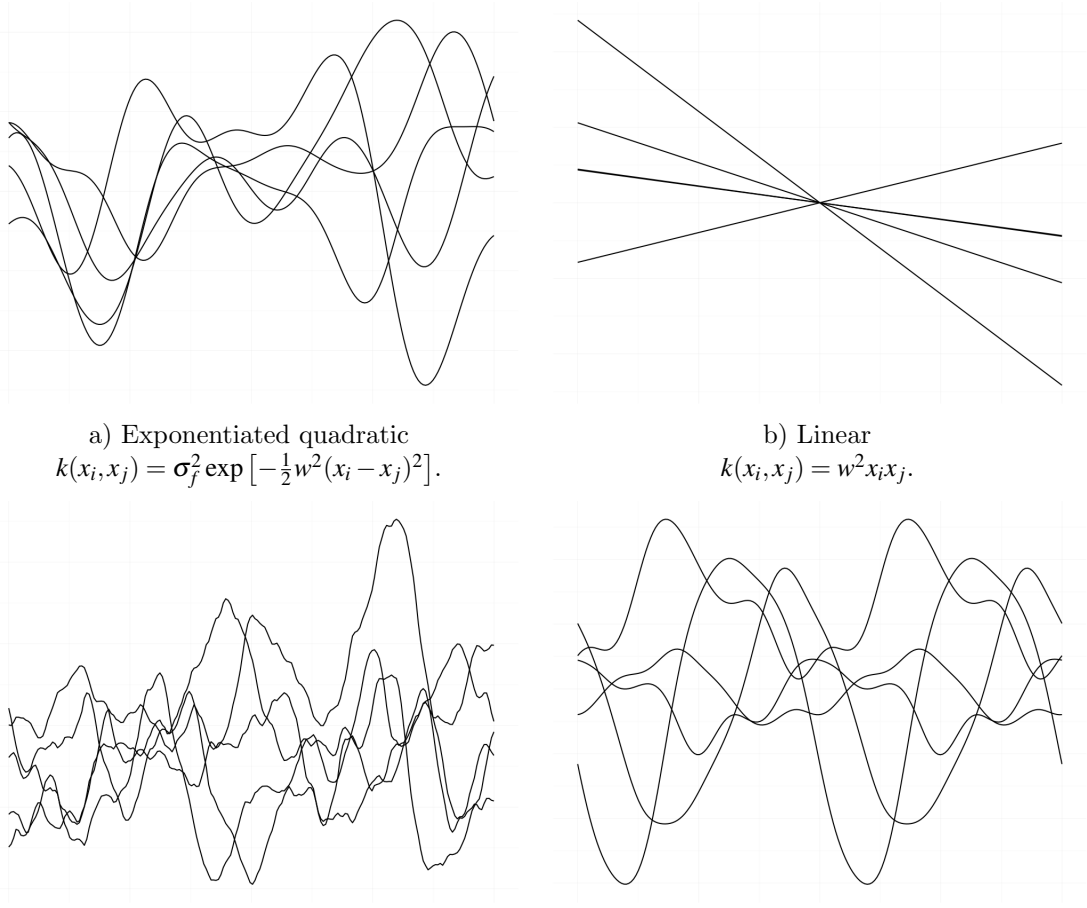
Any function that generates a positive semidefinite matrix $\boldsymbol{K}_f$ is a valid covariance function. Chapter 4 of the book by Rasmussen and Williams (2006) describes some common examples. Interestingly, the sum and/or product of any number of covariance functions, possibly scaled, is also a valid covariance function (SHAWE-TAYLOR; CRISTIANINI, 2004):

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_m A_m k_m(\boldsymbol{x}_i, \boldsymbol{x}_j) + \prod_n B_n k_n(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2.5}$$

where A_m and B_n are real positive constants. Furthermore, the following expression also results in valid covariance functions:

$$k_2(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(g(\boldsymbol{x}_i), g(\boldsymbol{x}_j)), \tag{2.6}$$

where $g(\cdot)$ is an arbitrary nonlinear mapping. Interestingly, the output dimension of $g(\boldsymbol{x}_i)$ can even be different from the original dimension of $\boldsymbol{x}_i$. All those properties turn easy the making of kernel matrices more suitable to a given dataset.



a) Exponentiated quadratic
$k(x_i,x_j) = \sigma_f^2 \exp\left[-\frac{1}{2}w^2(x_i-x_j)^2\right]$.

b) Linear
$k(x_i,x_j) = w^2 x_i x_j$.

c) Matérn 3/2
$k(x_i,x_j) = \sigma_f^2(1+\sqrt{3}w(x_i-x_j))\exp\left[-\sqrt{3}w(x_i-x_j)\right]$.

d) Periodic
$k(x_i,x_j) = \sigma_f^2 \exp\left[-2w^2 \sin^2\left(\frac{x_i-x_j}{2}\right)\right]$.

Figure 2 – Examples of GP samples with different covariance functions.

We can sample from the GP prior in Eq. (2.3) even before observing any data. Fig. 2 shows some samples for the unidimensional case using some of the covariance functions commonly used in the literature. We emphasize that a single sample defines an entire possible realization of the unknown function $f(\cdot)$.

Remark The kernel *hyperparameters* of the GP model (and other kernel-based approaches) are distinct from the *parameters* found, for instance, in neural networks. Instead of concentrating the knowledge extracted from the data, as the weights of a neural network, hyperparameters simply constrain and characterize the general behavior of the model. Moreover, they are usually present in a much lower number.

2.3 Inference from Noisy Observations

In practice, the true values of $\boldsymbol{f}$ are usually not available and the regression must be performed instead with the observations $y_i = f(\boldsymbol{x}_i) + \varepsilon_i^{(y)}, \forall i \in \{1,\cdots,N\}$, contaminated

with some noise $\varepsilon_i^{(y)}$. If we consider the observation noise to be independent and Gaussian with zero mean and variance σ_y^2, i.e., $\varepsilon_i^{(y)} \sim \mathcal{N}(0, \sigma_y^2)$, we obtain the *likelihood*

$$p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_y^2 \boldsymbol{I}), \tag{2.7}$$

where $\boldsymbol{y} \in \mathbb{R}^N$ is the vector of noisy observations and $\boldsymbol{I} \in \mathbb{R}^{N \times N}$ is the identity matrix. Since $\boldsymbol{f}$ is Gaussian (Eq. (2.3)), it can be marginalized analytically in order to obtain the *marginal likelihood*

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}) &= \int_{\boldsymbol{f}} p(\boldsymbol{y}|\boldsymbol{f}) p(\boldsymbol{f}|\boldsymbol{X}) \\ &= \int_{\boldsymbol{f}} \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_y^2 \boldsymbol{I}) \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}_f) \\ &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I}). \end{aligned} \tag{2.8}$$

Given a new input $\boldsymbol{x}_* \in \mathbb{R}^D$, the posterior predictive distribution of the related output $f_* \in \mathbb{R}$ is calculated analytically using standard Gaussian distribution conditioning properties (Eq. (2.2)):

$$\begin{aligned} p(f_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) &= \mathcal{N}\left(f_* \middle| \mu_*, \sigma_*^2\right), \\ \mu_* &= \boldsymbol{k}_{*f} (\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{y}, \\ \sigma_*^2 &= K_* - \boldsymbol{k}_{*f} (\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{k}_{f*}, \end{aligned} \tag{2.9}$$

which holds given the zero-mean joint Gaussian distribution below:

$$p(\boldsymbol{y}, f_*) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{y} \\ f_* \end{bmatrix} \middle| \boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I} & \boldsymbol{k}_{f*} \\ \boldsymbol{k}_{*f} & K_* \end{bmatrix} \right), \tag{2.10}$$

where $\boldsymbol{k}_{f*} = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}_*, \boldsymbol{x}_N)]^\top \in \mathbb{R}^N$, $\boldsymbol{k}_{*f} = \boldsymbol{k}_{f*}^\top$ and $K_* = k(\boldsymbol{x}_*, \boldsymbol{x}_*) \in \mathbb{R}$. The predictive distribution of the noisy version $y_* \in \mathbb{R}$ is simply given by $p(y_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) = \mathcal{N}\left(y_* \middle| \mu_*, \sigma_*^2 + \sigma_y^2\right)$. It is important to note that predictions done with Eq. (2.9) use all the estimation data $\boldsymbol{y}$. Furthermore, each prediction is a fully defined distribution, instead of a point estimate, which reflects the inherent uncertainty of the regression problem.

Fig. 3 illustrates a graphical model to represent the relations between the variables previously defined for the standard GP model for noisy regression. The observations are shown as filled nodes and the latent (unobserved) variables as white nodes. Since the inputs $\boldsymbol{x}_i$ are not random, they are shown as deterministic variables. The observation noise is made explicit by the Gaussian noise variables $\varepsilon_i^{(y)}$.

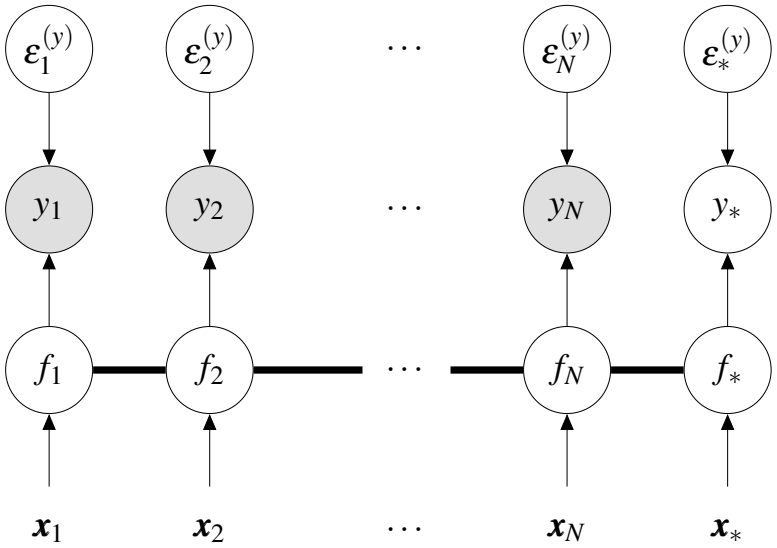


Figure 3 – Graphical model detailing the relations between the variables in a standard GP model. The observations are shown as filled nodes and the latent (unobserved) variables as white nodes. The thick bar that connects the latent variables f_i indicates that all the those variables are connected between themselves.

It is important to highlight some points in Fig. 3: (i) an observation y_i is conditionally independent of the other nodes given its associated latent function variable f_i and observation noise $\varepsilon_i^{(y)}$, which implies $p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{N} p(y_i|f_i)$; (ii) the thick bar that connects the latent variables f_i indicates that all the those variables are connected between themselves, which is expected from the definition of the covariance matrix $\boldsymbol{K}_f$ (see Eq. (2.4)); (iii) the prediction f_* is part of the same GP prior that models the N training samples, which justifies the joint distribution in Eq. (2.10).

Alternatively (and more extensively), we could obtain the predictive expressions in Eq. (2.9) following a more formal Bayesian approach by first calculating the posterior distribution of $\boldsymbol{f}$ given observed data via Bayes' rule, which is tractable[1] in this case:

$$\underbrace{p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})}_{\text{posterior}} = \frac{\overbrace{p(\boldsymbol{y}|\boldsymbol{f})}^{\text{likelihood}}\overbrace{p(\boldsymbol{f}|\boldsymbol{X})}^{\text{prior}}}{\underbrace{p(\boldsymbol{y}|\boldsymbol{X})}_{\text{marginal likelihood}}}$$

$$p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X}) \propto \mathcal{N}(\boldsymbol{y}|\boldsymbol{f},\sigma_y^2\boldsymbol{I})\mathcal{N}(\boldsymbol{f}|\boldsymbol{0},\boldsymbol{K}_f)$$

$$p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{K}_f(\boldsymbol{K}_f+\sigma_y^2\boldsymbol{I})^{-1}\boldsymbol{y}, (\boldsymbol{K}_f^{-1}+\sigma_y^{-2}\boldsymbol{I})^{-1}), \tag{2.11}$$

where the marginal likelihood acts as a normalization term and the final expression is obtained by working the product of the likelihood and the prior and then "completing the

[1] We use the terms *tractable* and *analytical* to refer to expressions that can be solved in closed form.

square"[2].

Now that we have both the likelihood $p(\boldsymbol{y}|\boldsymbol{f})$ and the posterior $p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})$, inference for a new output f_* given a new input $\boldsymbol{x}_*$ is obtained with the tractable integral below, which marginalizes (integrates out) the latent variable $\boldsymbol{f}$:

$$p(f_*|\boldsymbol{y},\boldsymbol{X},\boldsymbol{x}_*) = \int_{\boldsymbol{f}} p(f_*|\boldsymbol{f},\boldsymbol{X},\boldsymbol{x}_*)p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X}),$$

where $p(f_*|\boldsymbol{f},\boldsymbol{X},\boldsymbol{x}_*) = \mathcal{N}(f_*|\boldsymbol{k}_{*f}\boldsymbol{K}_f^{-1}\boldsymbol{t}, K_* - \boldsymbol{k}_{*f}\boldsymbol{K}_f^{-1}\boldsymbol{k}_{f*})$ (see Eq. (2.2)),

$$p(f_*|\boldsymbol{y},\boldsymbol{X},\boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2),$$

$$\mu_* = \boldsymbol{k}_{*f}(\boldsymbol{K}_f + \sigma_y^2\boldsymbol{I})^{-1}\boldsymbol{y},$$

$$\sigma_*^2 = K_* - \boldsymbol{k}_{*f}(\boldsymbol{K}_f + \sigma_y^2\boldsymbol{I})^{-1}\boldsymbol{k}_{f*},$$

which is equal to Eq. (2.9). We emphasize that many of the previous expressions are only tractable because of the GP prior and Gaussian likelihood. For instance, if the likelihood is not Gaussian, which implies non-Gaussian observations, inference would not be analytical anymore. In such case, as in Chapter 4 of this thesis, where we consider non-Gaussian heavy-tailed likelihoods, approximation methods are required.

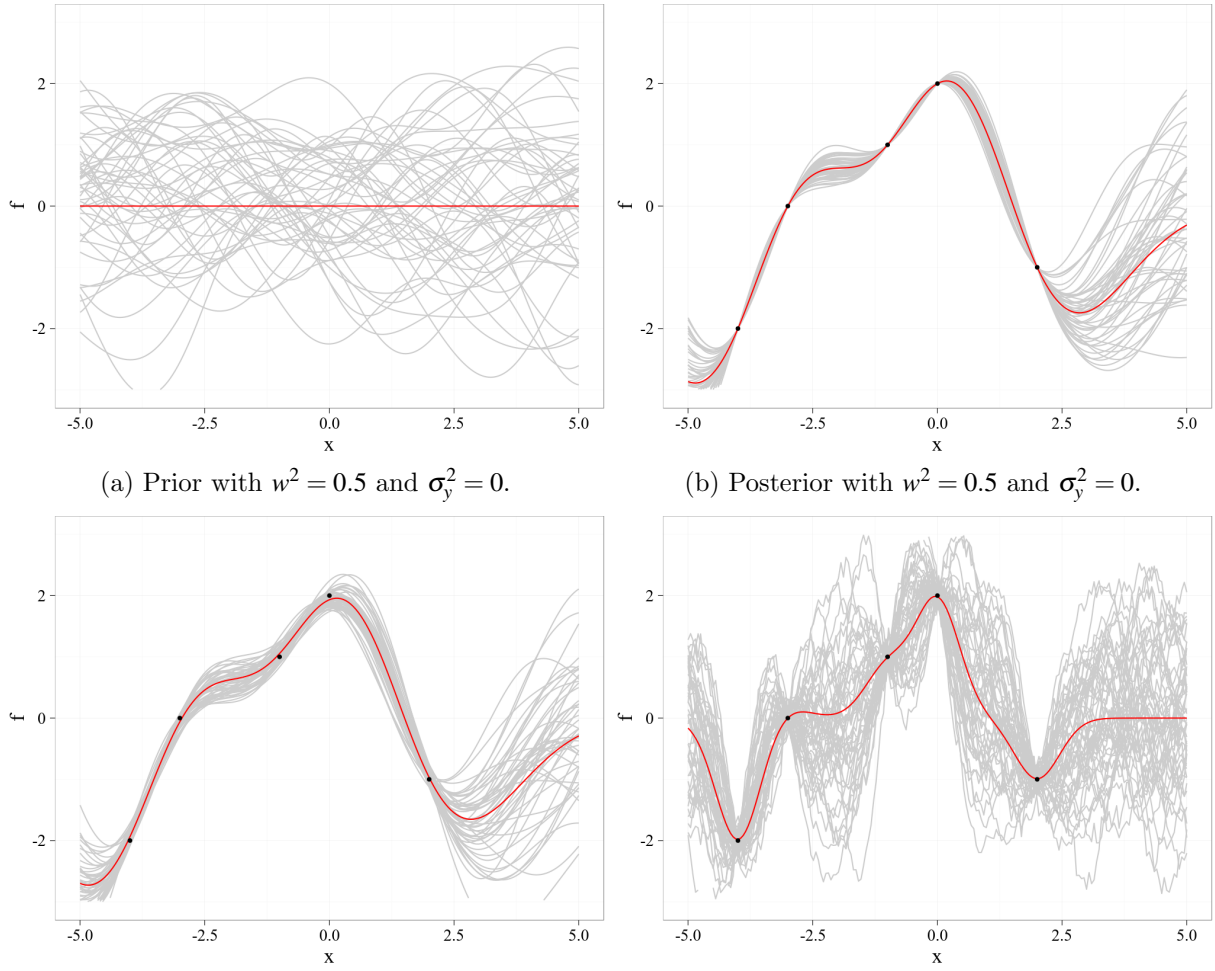
Fig. 4 shows *a priori* and *a posteriori* samples, i.e., before and after the observation of some outputs (represented by the small black dots). Note that larger values of w^2 (smaller lengthscales) are related to wigglier functions. Note also that when noisy outputs are considered ($\sigma_y^2 > 0$) the uncertainty around the observations is not close to zero anymore.

Remark The sampled curves shown in Fig. 4 illustrate a typical issue faced by nonlinear machine learning method: the finite (and possibly noisy) training set is responsible for the existence of multiple models that are able to explain the observations. Thus, one should be careful when choosing a single instance within a class of models and do so by following a clearly defined metric.

2.4 Bayesian Model Selection

Differently from parametric regression methods, GP models do not have parameters that concentrate the knowledge obtained from the training set. The only unknown

[2]This is a well known technique to obtain the moments of a Gaussian distribution from the quadratic form inside the exponential. See Bishop (2006), Section 2.3.1, for details.



(a) Prior with $w^2 = 0.5$ and $\sigma_y^2 = 0$.

(b) Posterior with $w^2 = 0.5$ and $\sigma_y^2 = 0$.

(c) Posterior with $w^2 = 0.5$ and $\sigma_y^2 = 0.01$.

(d) Posterior with $w^2 = 5$ and $\sigma_y^2 = 0.01$.

Figure 4 – Samples from the GP model before and after the observations. In all cases $\sigma_f^2 = 1$ was used.

model components are the kernel and noise hyperparameters. The noise variance σ_y^2 is usually included in the vector of kernel hyperparameters $\boldsymbol{\theta}$, whose length becomes $D+2$, a quantity often much smaller than, for example, the number of weights in a multilayer neural network.

The model selection step in the GP framework consists in finding hyperparameters that appropriately explain the available data. A common approach considers the maximization of the marginal log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})$ of the observed data with respect to the hyperparameters $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta}_{\text{ML}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})$, which is named a *maximum likelihood* solution. The so-called *evidence* of the model, which is obtained after

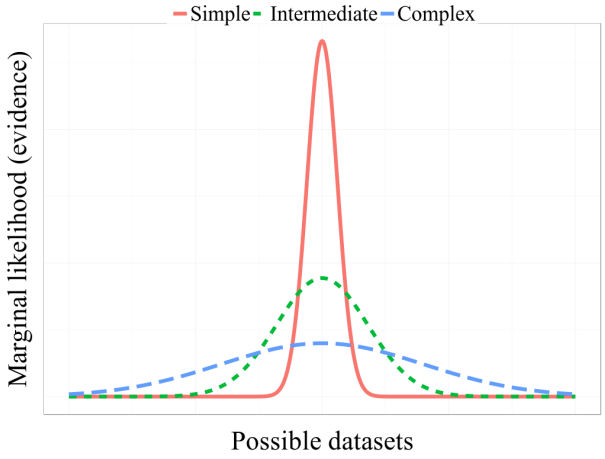


Figure 5 – Illustration of the Bayesian model selection procedure, where models too simple or too complex are avoided. The vertical axis indicates the evidence of a given model. Note that models with intermediate complexity present a balance between the value of the evidence and the number of possible supported datasets. Figure adapted from Rasmussen and Williams (2006).

marginalizing $\boldsymbol{f}$, is given by

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) &= \log \mathcal{N}(\boldsymbol{y}|\boldsymbol{0},\boldsymbol{K}_f+\sigma_y^2\boldsymbol{I}),\\ &= -\frac{N}{2}\log(2\pi) - \frac{1}{2}\underbrace{\log\left|\boldsymbol{K}_f+\sigma_y^2\boldsymbol{I}\right|}_{\text{model capacity}} - \frac{1}{2}\underbrace{\boldsymbol{y}^\top\left(\boldsymbol{K}_f+\sigma_y^2\boldsymbol{I}\right)^{-1}\boldsymbol{y}}_{\text{data fitting}},\end{aligned} \tag{2.12}$$

which follows the marginal likelihood derived in Eq. (2.8), but now we explicitly denote the dependency on the vector of hyperparameters $\boldsymbol{\theta}$, which is used to compute the covariance matrix $\boldsymbol{K}_f+\sigma_y^2\boldsymbol{I}$[3]. The *data fitting* term highlighted in Eq. (2.12) is the only one containing the observations $\boldsymbol{y}$. The other highlighted term is related to the *model capacity* and is equivalent to a complexity penalty. Thus, model selection by evidence maximization automatically balances between those two components. This procedure is also known as *type II maximum likelihood*, since the optimization is in the hyperparameter space, instead of the parameter space (RASMUSSEN; WILLIAMS, 2006). Fig. 5 illustrates such Bayesian selection methodology, where the preferred model should be an intermediate one, neither too simple nor too complex, which is able to efficiently explain a given dataset without being too restrictive or too generic.

The optimization of such model selection problem is guided by the analytical

[3]We emphasize that the Eq. (2.12) actually expresses the logarithm of a *likelihood function* of the hyperparameters $\boldsymbol{\theta}$ conditioned on the data and it is not a probability distribution (BISHOP, 2006).

Algorithm 1: Standard GP modeling for regression.

- Estimation step

Require: $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ (inputs), $\boldsymbol{y} \in \mathbb{R}^{N}$ (outputs)

Initialize model hyperparameters $\boldsymbol{\theta} = \left[\sigma_f^2, w_1^2, \cdots, w_D^2, \sigma_y^2\right]^\top$;

repeat

Compute the model evidence $\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})$ via Eq. (2.12).

Compute the analytical gradients $\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ via the Eq. (2.13);

Update $\boldsymbol{\theta}$ with a gradient-based method (e.g. BFGS (FLETCHER, 2013));

until convergence or maximum number or iterations

Output the optimized hyperparameters $\boldsymbol{\theta}_{ML}$;

- Test step

Require: $\boldsymbol{x}_* \in \mathbb{R}^{D}$ (inputs), $\boldsymbol{\theta}_{ML}$

Compute the predictive mean μ_* and variance σ_*^2 via the Eq. (2.9);

Output $y_* \sim \mathcal{N}\left(\mu_*, \sigma_*^2 + \sigma_y^2\right)$;

gradients of the evidence with respect to each component of $\boldsymbol{\theta}$:

$$\begin{aligned}\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = -\frac{1}{2}\operatorname{Tr}\left((\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})^{-1}\frac{\partial (\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})}{\partial \boldsymbol{\theta}}\right) \\ + \frac{1}{2}\boldsymbol{y}^\top (\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})^{-1}\frac{\partial (\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})}{\partial \boldsymbol{\theta}}(\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})^{-1}\boldsymbol{y}.\end{aligned} \tag{2.13}$$

After the optimization, σ_f^2 is proportional to the overall variance of the output, while σ_y^2 becomes closer to the observation noise variance. The optimized ARD hyperparameters $w_1^2, \ldots, w_D^2$ are able to automatically turn off unnecessary dimensions of the input by taking values close to zero. Importantly, the model selection procedure does not involve any grid or random search, mechanisms usually needed for other kernel methods. Algorithm 1 summarizes the use of the standard GP modeling framework for regression.

Remark It should be noted that the aforementioned maximum likelihood solution for the kernel hyperparameters, although by far the most used in practice, is not the only possible approach to perform model selection with standard GP models. For instance, we could choose prior distributions for the components of the vector $\boldsymbol{\theta}$ and find a *maximum a posteriori* (MAP) solution, i.e., $\boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})$. Rasmussen and Williams (2006) also describe cross-validation methodologies, very common in other machine learning methods. Another alternative is to consider a marginalization of the hyperparameters themselves, which although non-analytical, can be performed approximately via sampling techniques (OSBORNE *et al.*, 2008; SVENSSON *et al.*, 2015).

2.5 From Feature Spaces to GPs

Despite the easiness of explaining the GP modeling approach from the function space view, as in Sections 2.2 and 2.3, it is useful to present the same expressions derived in a different manner.

Let $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^V$ be a feature mapping function, from a D-dimensional space to a V-dimensional space, and $\boldsymbol{w} \in \mathbb{R}^V$ a vector of weights (or parameters). Using the same notation for inputs and outputs of previous sections, $\boldsymbol{w}$ can be used in the standard Bayesian linear (in the parameters) regression model below:

$$y_i = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}_i) + \varepsilon_i^{(y)}, \quad \varepsilon_i^{(y)} \sim \mathcal{N}(0, \sigma_y^2). \tag{2.14}$$

If we consider a zero mean Gaussian prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \boldsymbol{\Sigma}_w)$ over the weights, with covariance matrix $\boldsymbol{\Sigma}_w \in \mathbb{R}^{V \times V}$, and define $\boldsymbol{\Phi} = \boldsymbol{\phi}(\boldsymbol{X}) \in \mathbb{R}^{V \times N}$ as a matrix where each column is given by $\boldsymbol{\phi}(\boldsymbol{x}_i)$, after the application of the Bayes' rule we get the posterior

$$p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) = \frac{p(\boldsymbol{y}|\boldsymbol{w}, \boldsymbol{X}) p(\boldsymbol{w})}{p(\boldsymbol{y}|\boldsymbol{X})} = \mathcal{N}\left(\boldsymbol{w} \middle| \frac{1}{\sigma_y^2} \boldsymbol{A}^{-1} \boldsymbol{\Phi} \boldsymbol{y}, \boldsymbol{A}^{-1}\right), \tag{2.15}$$

where $\boldsymbol{A} \in \mathbb{R}^{V \times V} = \frac{1}{\sigma_y^2} \boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \boldsymbol{\Sigma}_w^{-1}$.

Given a new input $\boldsymbol{x}_*$, prediction is performed by averaging the weights, i.e., integrating out $\boldsymbol{w}$:

$$\begin{aligned} p(f_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) &= \int_{\boldsymbol{w}} p(f_*|\boldsymbol{x}_*, \boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{y}, \boldsymbol{X}) \\ &= \mathcal{N}\left(f_* \middle| \frac{1}{\sigma_y^2} \boldsymbol{\phi}(\boldsymbol{x}_*)^\top \boldsymbol{A}^{-1} \boldsymbol{\Phi} \boldsymbol{y}, \boldsymbol{\phi}(\boldsymbol{x}_*)^\top \boldsymbol{A}^{-1} \boldsymbol{\phi}(\boldsymbol{x}_*)\right). \end{aligned} \tag{2.16}$$

We can rewrite the predictive distribution using the definition of the matrix $\boldsymbol{A}$ and the two matrix identities below:

$$\begin{aligned} \boldsymbol{A}^{-1} \boldsymbol{\Phi} &= \left(\frac{1}{\sigma_y^2} \boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \boldsymbol{\Sigma}_w^{-1}\right)^{-1} \boldsymbol{\Phi} = \sigma_y^2 \boldsymbol{\Sigma}_w \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma_y^2 \boldsymbol{I}\right)^{-1}, \\ \boldsymbol{A}^{-1} &= \left(\frac{1}{\sigma_y^2} \boldsymbol{\Phi} \boldsymbol{\Phi}^\top + \boldsymbol{\Sigma}_w^{-1}\right)^{-1} = \boldsymbol{\Sigma}_w - \boldsymbol{\Sigma}_w \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma_y^2 \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w, \end{aligned}$$

where the second expression is the so-called *matrix inversion lemma*. The first identity is directly used in the mean of Eq. (2.16), while the lemma is applied in the variance. The new prediction then becomes

$$\begin{aligned} p(f_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}\Big(f_* \Big| &\boldsymbol{\phi}(\boldsymbol{x}_*)^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} (\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{y}, \\ &\boldsymbol{\phi}(\boldsymbol{x}_*)^\top \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\boldsymbol{x}_*) - \boldsymbol{\phi}(\boldsymbol{x}_*)^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} (\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\boldsymbol{x}_*)\Big). \end{aligned} \tag{2.17}$$

We can define some quantities by applying the *kernel trick* and the definition of a kernel function (SMOLA; SCHÖLKOPF, 2002):

$$\begin{aligned}
\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} &= \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w^{1/2} \boldsymbol{\Sigma}_w^{1/2} \boldsymbol{\Phi} = \boldsymbol{\Omega}^\top \boldsymbol{\Omega} = k(\boldsymbol{X}, \boldsymbol{X}) = \boldsymbol{K}_f, \\
\boldsymbol{\phi}(\boldsymbol{x}_*)^\top \boldsymbol{\Sigma}_w \boldsymbol{\Phi} &= k(\boldsymbol{x}_*, \boldsymbol{X}) = \boldsymbol{k}_{*f}, \\
\boldsymbol{\Phi}^\top \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\boldsymbol{x}_*) &= k(\boldsymbol{X}, \boldsymbol{x}_*) = \boldsymbol{k}_{f*}, \\
\boldsymbol{\phi}(\boldsymbol{x}_*)^\top \boldsymbol{\Sigma}_w \boldsymbol{\phi}(\boldsymbol{x}_*) &= k(\boldsymbol{x}_*, \boldsymbol{x}_*) = K_*,
\end{aligned}$$

where we have denoted $\boldsymbol{\Omega}$ as a modified feature map obtained from the input and replaced the inner product of such mapping by the kernel matrix $\boldsymbol{K}_f$. It is important to emphasize that the actual mapping is never performed explicitly, but only by using the kernel function.

Finally, by replacing the new kernel expressions back in Eq. (2.17), we get the standard predictive expression for GP models:

$$p(f_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{k}_{*f}(\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{y}, K_* - \boldsymbol{k}_{*f}(\boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I})^{-1} \boldsymbol{k}_{f*}). \tag{2.18}$$

Now we can see that GP regression is related to Bayesian regression with basis functions, i.e., linear combination of possibly nonlinear mappings, where a Gaussian prior is given to the weights, which are analytically integrated out. We emphasize that such integration is equivalent to consider infinitely many weights pondered by their priors. Furthermore, it is well known from the kernel learning literature, that many kernel functions, such as the previously mentioned exponentiated quadratic (Eq. (2.4)) is related to an infinite-dimensional feature space $\boldsymbol{\Omega}$ (SMOLA; SCHÖLKOPF, 2002).

Perhaps even more interestingly, in the chapter entitled *Priors for Infinite Networks* in his thesis (NEAL, 1994), Neal demonstrated that within the Bayesian formalism, neural networks with one hidden layer containing infinitely many hidden neurons converge to a GP when Gaussian priors are assigned to the neurons' weights. Furthermore, he also stated that such model should be able to avoid the risk of overfitting. Later, Williams (1998) derived the covariance function related to such infinity limit.

Remark Rasmussen and Williams (2006) dedicate a whole chapter in their book to the relations between GPs and other learning methods, such as Reproducing Kernel Hilbert Spaces (RKHS), regularizers, spline models, support vector machines (SVM) and relevance vector machines (RVM), emphasizing similarities between each framework. For instance, it is possible to interpret SVMs as MAP solutions to a GP inference problem (SOLLICH,

2002). Connections with kernel extensions to classical adaptive filters, such as the kernel recursive least-squares (KRLS) (ENGEL *et al.*, 2004) and kernel least mean squares (KLMS) (LIU *et al.*, 2008) algorithms have also been studied by Vaerenbergh *et al.* (2012) and Vaerenbergh *et al.* (2016), respectively.

2.6 Sparse GP Models

A problem usually associated with GP models is its $\mathcal{O}(N^3)$ computation complexity and $\mathcal{O}(N^2)$ memory requirement[4], related to the predictive expression in Eq. (2.9) and the evidence in Eq. (2.12). The latter is more critical, since along with the gradients in Eq. (2.13), it needs to be computed every iteration of the optimization procedure. When the number of samples N is larger than a few thousands, the matrix inverse and determinant operations can be slow or even prohibitive.

Several authors have proposed different solutions to deal with this problem. Most of such methods were covered by Quiñonero-Candela and Rasmussen (2005), where a unifying view for sparse GP modeling is presented, highlighting the implicit approximations considered by each approach.

Later, a sparse approximate GP framework was proposed by Titsias (2009a), following a variational Bayes approach (SCHWARZ, 1988; JORDAN *et al.*, 1999; GIBBS; MACKAY, 2000). Such approach has been proven to be flexible and effective by other authors, for instance in the recent works by Matthews *et al.* (2016) and Bauer *et al.* (2016). Since Titsias' variational sparse framework, sometimes called the variational *free energy* approximation, is applied within several models addressed by this thesis, we will describe it in more detail here. We follow the original presentation by Titsias (2009a) and the one presented by Damianou (2015). The reader is referred to Quiñonero-Candela and Rasmussen (2005) and Rasmussen and Williams (2006) to learn more about other sparse GP approaches.

2.6.1 The Variational Sparse GP Framework

The standard GP evidence expression in Eq. (2.12) depends on the inversion of the matrix $\boldsymbol{K}_f$, which scales cubicly with N. In order to avoid that, Titsias' approach

[4]The cubic computational complexity and squared memory demand come, respectively, from the expensive inverse of the kernel matrix $\boldsymbol{K}_f$ and its storage.

starts by augmenting the model and including M samples $\boldsymbol{z} \in \mathbb{R}^M$ from the same GP prior of the vector $\boldsymbol{f}$. Those samples, named *inducing points*, are related to M *inducing inputs*, or *pseudo-inputs*, $\boldsymbol{\zeta}_j|_{j=1}^M \in \mathbb{R}^D$ that live in the same space of the inputs $\boldsymbol{x}_i|_{i=1}^N$. Thus, the following joint distribution holds:

$$p(\boldsymbol{f},\boldsymbol{z}) = \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{f} \\ \boldsymbol{z} \end{array}\right] \middle| \boldsymbol{0}, \left[\begin{array}{cc} \boldsymbol{K}_f & \boldsymbol{K}_{fz} \\ \boldsymbol{K}_{fz}^\top & \boldsymbol{K}_z \end{array}\right]\right), \tag{2.19}$$

where the elements of the covariance matrices $\boldsymbol{K}_z \in \mathbb{R}^{M\times M}$ and $\boldsymbol{K}_{fz} \in \mathbb{R}^{N\times M}$ are given respectively by $[\boldsymbol{K}_z]_{jj'} = k(\boldsymbol{\zeta}_j, \boldsymbol{\zeta}_{j'})$ and $[\boldsymbol{K}_{fz}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{\zeta}_j)$.

The new marginal likelihood $p(\boldsymbol{y}|\boldsymbol{X})$ then becomes

$$p(\boldsymbol{y}|\boldsymbol{X}) = \int_{\boldsymbol{f},\boldsymbol{z}} p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X})p(\boldsymbol{z}), \tag{2.20}$$

where $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{K}_z)$ and the conditioning on the pseudo-inputs $\boldsymbol{\zeta}_j$ is made implicit by the presence of the covariance matrix $\boldsymbol{K}_z$, which is calculated from them. The dependence on the kernel hyperparameters $\boldsymbol{\theta}$ in Eq. (2.20) was omitted and the subindexes in the integral symbol indicate the integrated variables, considering all the possible values they can take. So far we have not really altered the original model, since if we integrate out $\boldsymbol{z}$, i.e., marginalize it, we recover the exact marginal likelihood of the standard GP model.

From the joint distribution in Eq. (2.19) and the Gaussian conditioning property, we know that

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X}) &= \mathcal{N}(\boldsymbol{f}|\boldsymbol{a}_f,\boldsymbol{\Sigma}_f), \\ \boldsymbol{a}_f &= \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{z}, \\ \boldsymbol{\Sigma}_f &= \boldsymbol{K}_f - \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{K}_{fz}^\top. \end{aligned} \tag{2.21}$$

It is important to note that, differently from the original distribution $p(\boldsymbol{f}|\boldsymbol{X})$ from standard GP, in Eq. (2.21), we do not need to invert the full matrix $\boldsymbol{K}_f$, but only the sparse matrix $\boldsymbol{K}_z$. Thus, when choosing $M < N$ we already get better performance for large N values, since the computational complexity becomes $\mathcal{O}(NM^2)$, due to the most expensive operation being now the matrix products in Eq. (2.21).

We proceed by following the standard variational approach (JORDAN *et al.*, 1999), denoting Q as a generic variational distribution and multiplying the right side of Eq. (2.20) by $\frac{Q}{Q}$. After applying Jensen's inequality[5], we are able to obtain a lower bound

[5]In its probabilistic form, Jensen's inequality states that $\phi(\mathbb{E}\{x\}) \leq \mathbb{E}\{\phi(x)\}$, where $\phi(\cdot)$ is a convex function and $\mathbb{E}\{\cdot\}$ is the expectation operator. Since the function $\log(\cdot)$ is concave, the side of the inequality in Eq. (2.22) is changed.

for the marginal log-likelihood:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \int_{\boldsymbol{f},\boldsymbol{z}} Q \log \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X})p(\boldsymbol{z})}{Q}. \tag{2.22}$$

The bound in Eq. (2.22) holds for any valid distribution Q. Conveniently, we choose the form $Q = q(\boldsymbol{z})p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X})$, which enable us to cancel the term $p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X})$ in the numerator inside the logarithm. The bound then becomes:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \int_{\boldsymbol{f},\boldsymbol{z}} q(\boldsymbol{z})p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X}) \log \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{z})}{q(\boldsymbol{z})},$$

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \int_{\boldsymbol{z}} q(\boldsymbol{z}) \Bigg\{ \underbrace{\int_{\boldsymbol{f}} p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X}) \log p(\boldsymbol{y}|\boldsymbol{f})}_{\mathscr{L}_1} + \log \frac{p(\boldsymbol{z})}{q(\boldsymbol{z})} \Bigg\}. \tag{2.23}$$

Now we integrate over $\boldsymbol{f}$ in Eq. (2.23), using the distribution in Eq. (2.21):

$$\begin{aligned}
\mathscr{L}_1 &= \int_{\boldsymbol{f}} p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X}) \log p(\boldsymbol{y}|\boldsymbol{f}) \\
&= \int_{\boldsymbol{f}} \mathscr{N}(\boldsymbol{f}|\boldsymbol{a}_f, \boldsymbol{\Sigma}_f) \log \mathscr{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_y^2 \boldsymbol{I}) \\
&= -\frac{N}{2} \log 2\pi\sigma_y^2 - \frac{1}{2\sigma_y^2} \left(\boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{y}^\top \boldsymbol{a}_f + \left(\boldsymbol{a}_f\right)^\top \boldsymbol{a}_f + \mathrm{Tr}(\boldsymbol{\Sigma}_f) \right) \\
&= \log \mathscr{N}(\boldsymbol{y}|\boldsymbol{a}_f, \sigma_y^2 \boldsymbol{I}) - \frac{1}{2\sigma_y^2} \mathrm{Tr}(\boldsymbol{\Sigma}_f).
\end{aligned}$$

Replacing the result of the latter integral in the original bound in Eq. (2.23) we get:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log \frac{\mathscr{N}(\boldsymbol{y}|\boldsymbol{a}_f, \sigma_y^2 \boldsymbol{I}) p(\boldsymbol{z})}{q(\boldsymbol{z})} - \frac{1}{2\sigma_y^2} \mathrm{Tr}(\boldsymbol{\Sigma}_f). \tag{2.24}$$

The integral in the last expression has a format similar to the first Jensen's inequality used in Eq. (2.22). If we revert such inequality, i.e., moving the logarithm outside the integral, we are able to optimally remove the dependency on $q(\boldsymbol{z})$ and obtain a tighter bound:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \log \int_{\boldsymbol{z}} \mathscr{N}(\boldsymbol{y}|\boldsymbol{a}_f, \sigma_y^2 \boldsymbol{I}) \mathscr{N}(\boldsymbol{z}|\boldsymbol{0}, \boldsymbol{K}_z) - \frac{1}{2\sigma_y^2} \mathrm{Tr}(\boldsymbol{\Sigma}_f)$$

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \log \mathscr{N}(\boldsymbol{y}|\boldsymbol{0}, \sigma_y^2 \boldsymbol{I} + \boldsymbol{K}_{fz} \boldsymbol{K}_z^{-1} \boldsymbol{K}_{fz}^\top) - \frac{1}{2\sigma_y^2} \mathrm{Tr}(\boldsymbol{K}_f - \boldsymbol{K}_{fz} \boldsymbol{K}_z^{-1} \boldsymbol{K}_{fz}^\top), \tag{2.25}$$

where we have used the following Gaussian integral identity:

$$p(\boldsymbol{a}|\boldsymbol{b}) = \mathscr{N}(\boldsymbol{a}|\boldsymbol{A}\boldsymbol{b} + \boldsymbol{m}, \boldsymbol{\Sigma}),$$

$$\text{where } p(\boldsymbol{b}) = \mathscr{N}(\boldsymbol{b}|\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b),$$

$$\text{and } p(\boldsymbol{a}) = \int_{\boldsymbol{b}} p(\boldsymbol{a}|\boldsymbol{b}) p(\boldsymbol{b}) = \mathscr{N}(\boldsymbol{a}|\boldsymbol{A}\boldsymbol{\mu}_b + \boldsymbol{m}, \boldsymbol{\Sigma} + \boldsymbol{A}\boldsymbol{\Sigma}_b \boldsymbol{A}^\top).$$

The final sparse GP variational bound in Eq. (2.25) can be used as a proxy for the true marginal log-likelihood $p(\boldsymbol{y}|\boldsymbol{X})$. Since it depends only on the trace of the possibly large matrix $\boldsymbol{K}_f$, it also causes the memory requirement to be proportional to $\mathcal{O}(NM)$, the size of the matrix $\boldsymbol{K}_{fz}$, lower than the required $\mathcal{O}(N^2)$ for standard GP when $M < N$.

The kernel hyperparameters and the additional variational parameters, i.e., the pseudo-inputs $\boldsymbol{\zeta}_j|_{j=1}^M$, can be optimized by maximizing the bound using its analytical gradients, making it closer to the true model evidence. Titsias (2009a) emphasizes how the pseudo-inputs are not model parameters, but variational parameters related to the chosen approximate inference method and, hence, their optimization do not cause overfitting. It is also stated that more pseudo-inputs can only make the bound tighter and improve the approximation[6]. Indeed, if we choose $M = N$ and turn the pseudo-inputs equal to the real inputs, we would recover the original GP model.

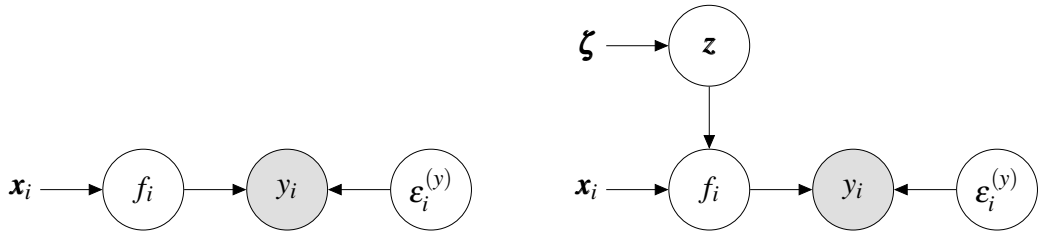
It is important to highlight that the way the variational sparse bound was derived, via Jensen's inequality in Eq. (2.22), was chosen for mathematical convenience. A more rigorous approach consists in directly consider the Kullback-Leibler (KL) divergence between the variational posterior Q and the true posterior $p(\boldsymbol{f},\boldsymbol{z}|\boldsymbol{y},\boldsymbol{X})$, as follows (BLEI *et al.*, 2017):

$$\begin{aligned}
\mathrm{KL}(Q||p(\boldsymbol{f},\boldsymbol{z}|\boldsymbol{y},\boldsymbol{X})) &= \int_{\boldsymbol{f},\boldsymbol{z}} Q \log \frac{Q}{p(\boldsymbol{f},\boldsymbol{z}|\boldsymbol{y},\boldsymbol{X})} \\
&= \int_{\boldsymbol{f},\boldsymbol{z}} Q \log \frac{Q p(\boldsymbol{y}|\boldsymbol{X})}{p(\boldsymbol{f},\boldsymbol{z},\boldsymbol{y},\boldsymbol{X})} \\
&= -\int_{\boldsymbol{f},\boldsymbol{z}} Q \log \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{z},\boldsymbol{X})p(\boldsymbol{z})}{Q} + \log p(\boldsymbol{y}|\boldsymbol{X}),
\end{aligned} \tag{2.26}$$

where we have used $p(\boldsymbol{f},\boldsymbol{z}|\boldsymbol{y},\boldsymbol{X}) = \frac{p(\boldsymbol{f},\boldsymbol{z},\boldsymbol{y},\boldsymbol{X})}{p(\boldsymbol{y}|\boldsymbol{X})}$. Since the KL divergence is always non-negative (Gibbs' inequality), we recover the inequality expressed in Eq. (2.22). Moreover, we can see from Eq. (2.26) that the maximization of the derived lower bound, i.e., the integral in the right side, is equivalent to minimizing the divergence between the approximation and the true posterior.

Predictions in the variational sparse framework can be done by applying the

[6]It is worth noting that, in practice, the use of a very large number of pseudo-inputs can turn the optimization slow and difficult.



(a) Standard GP. (b) Sparse GP augmented with inducing points.

Figure 6 – Simplified graphical models for the standard and augmented sparse GPs.

following expressions:

$$p(f_*|\boldsymbol{y},\boldsymbol{X},\boldsymbol{x}_*) = \mathcal{N}\left(f_*\middle|\mu_*,\sigma_*^2\right), \tag{2.27}$$
$$\mu_* = \boldsymbol{k}_{*z}\boldsymbol{K}_z^{-1}\boldsymbol{m},$$
$$\sigma_*^2 = K_* - \boldsymbol{k}_{*z}\boldsymbol{K}_z^{-1}\boldsymbol{k}_{z*} + \boldsymbol{k}_{*z}\boldsymbol{K}_z^{-1}\boldsymbol{S}\boldsymbol{K}_z^{-1}\boldsymbol{k}_{z*},$$

where $\boldsymbol{m}$ and $\boldsymbol{S}$ are the moments of the optimal variational distribution $q^*(\boldsymbol{z})$, given by

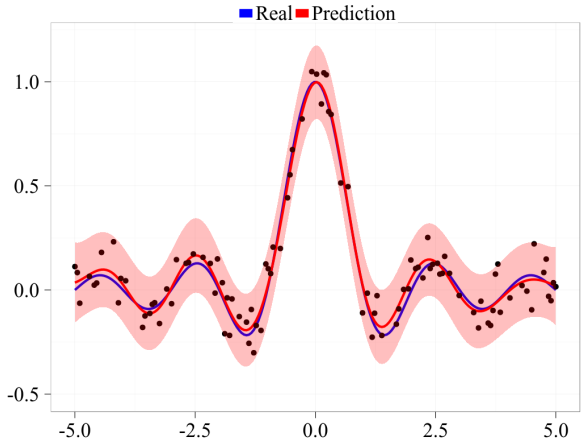
$$q^*(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{m},\boldsymbol{S}) \tag{2.28}$$
$$\boldsymbol{m} = \frac{1}{\sigma_y^2}\boldsymbol{K}_z\left(\boldsymbol{K}_z + \frac{1}{\sigma_y^2}\boldsymbol{K}_{fz}^\top\boldsymbol{K}_{fz}\right)^{-1}\boldsymbol{K}_{fz}^\top\boldsymbol{y} \tag{2.29}$$
$$\boldsymbol{S} = \boldsymbol{K}_z\left(\boldsymbol{K}_z + \frac{1}{\sigma_y^2}\boldsymbol{K}_{fz}^\top\boldsymbol{K}_{fz}\right)^{-1}\boldsymbol{K}_z. \tag{2.30}$$

Such optimal distribution is derived in the Appendix A.2, though in a slightly different context.

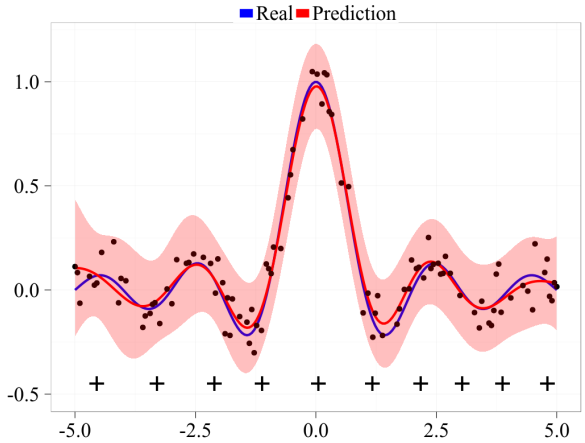
Fig. 6 shows simplified graphical models[7] that illustrate how the sparse GP model augmented with inducing points $\boldsymbol{z}$ compares with standard GP. White nodes are related to latent (unobserved) probabilistic variables, while filled nodes are the random observations. The illustrations explicit the noise model, represented by the observational Gaussian noise $\varepsilon_i^{(y)}$, and the latent function values f_i, even though the latter are always analytically integrated out.

A simple regression example is illustrated in Fig. 7, in order to compare the standard GP model and the variational sparse approximation. The models were trained with 100 samples from the normalized *sinc* function ($f_i = \frac{\sin(\pi x_i)}{\pi x_i}$) contaminated with noise sampled from $\mathcal{N}(0, 0.005)$. Two sparse models were trained, one with only $M = 10$

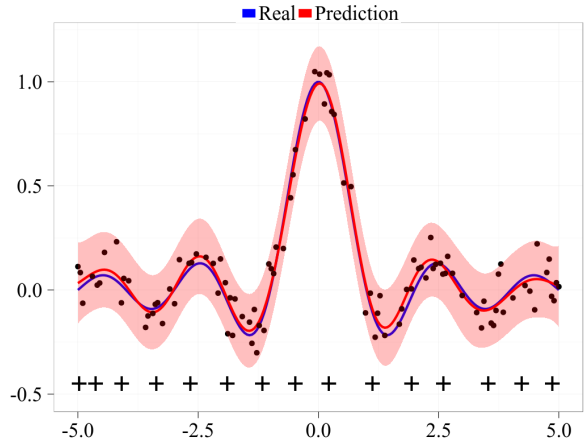
[7] For the sake of simplicity, most of the graphical models in this thesis will show only the dependencies between the variables related to the i-th observation, which is equivalent to a single column of the full GP diagram presented in Fig. 3.



(a) Standard GP.



(b) Variational sparse GP ($M = 10$).



c) Variational sparse GP ($M = 20$).

Figure 7 – Comparison of standard GP and variational sparse GP regression. The black dots are the training samples, while the small crosses indicate the position of the optimized pseudo-inputs. Note that in the third scenario, for $M = 20$, some pseudo-inputs were optimized to locations outside the plot area.

pseudo-inputs and other with $M = 20$. By looking to the predicted curves, we can see that the sparse variant with more pseudo-inputs is very similar to the full GP model. Even though the version with $M = 10$ seems a bit worse, it could be just enough, for instance, in applications with computational restrictions.

Remark Matthews *et al.* (2016) argue against the "augmentation" interpretation behind original Titsias' presentation of the variational sparse GP framework and present a more precise and rigorous treatment, mentioning that it leads to the same results. Nevertheless, we maintained the original aforementioned derivation due to its simpler intuition and common presence in the literature.

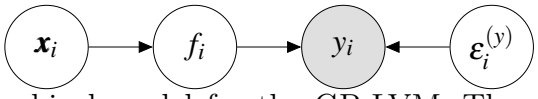


Figure 8 – Simplified graphical model for the GP-LVM. The main difference from the standard supervised GP model is that the inputs are now latent and the only observed variables are the noisy observations.

2.7 Unsupervised GP Modeling and Uncertain Inputs

So far we have only referred to supervised GP modeling, i.e., the mapping from a set of inputs to a set of correspondent outputs, all of them observed. However, GP models have also been applied to unsupervised tasks, where only a dataset $\boldsymbol{Y} \in \mathbb{R}^{N \times D_y}$ comprised of N noisy D_y-dimensional observations is available.

The Gaussian Process Latent Variable Model (GP-LVM), proposed by Lawrence (2004), aims to tackle such class of problems. It considers that the data $\boldsymbol{Y}$ is generated by transforming a set of latent variables $\boldsymbol{X} \in \mathbb{R}^{N \times D_x}$ by a GP. Thus, the GP-LVM can be summarized by the set of distributions below:

$$p(\boldsymbol{F}|\boldsymbol{X}) = \prod_{d=1}^{D_y} \mathcal{N}(\boldsymbol{f}_{:d}|\boldsymbol{0}, \boldsymbol{K}_f), \tag{2.31}$$

$$p(\boldsymbol{Y}|\boldsymbol{F}, \boldsymbol{X}) = \prod_{d=1}^{D_y} \mathcal{N}(\boldsymbol{y}_{:d}|\boldsymbol{f}_{:d}, \sigma_y^2 \boldsymbol{I}) \mathcal{N}(\boldsymbol{f}_{:d}|\boldsymbol{0}, \boldsymbol{K}_f), \tag{2.32}$$

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{d=1}^{D_y} \mathcal{N}(\boldsymbol{y}_{:d}|\boldsymbol{0}, \boldsymbol{K}_f + \sigma_y^2 \boldsymbol{I}), \tag{2.33}$$

where $\boldsymbol{F} \in \mathbb{R}^{N \times D_y}$ is the latent noiseless version of $\boldsymbol{Y}$, the covariance matrix $\boldsymbol{K}_f \in \mathbb{R}^{N \times N}$ is computed from $\boldsymbol{X}$, σ_y^2 is the noise variance and each output dimension d is modeled by a separate GP prior, although with the same kernel hyperparameters in this example. For the sake of simplicity we have also considered the same noise variance for all the output dimensions. We used the notation $\boldsymbol{y}_{:d} \in \mathbb{R}^N$ to denote the vector comprised of the d-th component of each observed sample, i.e., $\boldsymbol{y}_{:d}$ is formed by the elements $Y_{id}|_{i=1}^N$. Note that, following the standard GP modeling framework, we were able to analytically integrate the latent function values $\boldsymbol{f}_{:d} \in \mathbb{R}^N$, comprised of the elements $F_{id}|_{i=1}^N$, to obtain Eq. (2.33). Fig. 8 illustrates the GP-LVM graphical model for one-dimensional ($D_y = 1$) observations.

The model described by former expressions differ from standard GP regression because the model inputs are not observed. Thus, they should be marginalized (integrated

out), like the latent function values $\boldsymbol{F}$. However, such marginalization is intractable[8], since the latent variables $\boldsymbol{X}$ appear in a complicated way inside the covariance matrix $\boldsymbol{K}_f$ in the Eq. (2.33).

The approach chosen by Lawrence (2005) to perform inference with the GP-LVM model consists in putting a Gaussian prior to the latent space and finding the MAP solution for the latent variables $\boldsymbol{X}$ as follows:

$$\boldsymbol{X}_{\text{MAP}} = \arg\max_{\boldsymbol{X}} \log p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X}), \tag{2.34}$$

where $p(\boldsymbol{X}) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{0},\boldsymbol{I})$.

The same expression in Eq. (2.34) can be used to jointly optimize the latent variables as well as the kernel hyperparameters via its analytical gradients.

The GP-LVM framework was originally proposed in the context of nonlinear dimensionality reduction[9], which can be done by simply choosing $D_x < D_y$, which results in finding a lower dimensional space to represent the original data. However, the approach has shown to be flexible enough to be used in several other scenarios. For instance, in supervised tasks, the matrix $\boldsymbol{X}$ can be seen as a set of uncertain inputs (DAMIANOU *et al.*, 2016). Besides, by manipulating the prior $p(\boldsymbol{X})$ on the latent variables, the GP-LVM can be further extended. We will see in Chapter 3 that many GP approaches for dynamical modeling are derived from a GP-LVM with dynamical priors.

Although tractable, the original MAP solution for the GP-LVM has some drawbacks. First, since it directly optimizes the latent variables, it is susceptible to overfitting. Second, the increase of the latent space dimension, i.e., larger D_x values, always result in better fit to the training data, which turns the optimization of the latent space dimensionality infeasible.

The Bayesian GP-LVM proposed by Titsias and Lawrence (2010) tackles the aforementioned issues by applying a variational approach in order to approximately integrate the model latent variables $\boldsymbol{X}$. Inspired by Titsias' variational sparse GP framework (TITSIAS, 2009a), the Bayesian GP-LVM guards against overfitting by considering the uncertainty of the latent space and enables automatic determination of D_x by using a

[8] An *intractable* expression is non-analytical, i.e., cannot be solved in closed form, and require some kind of approximate solution.

[9] The GP-LVM can be seen as a nonlinear extension of the regular probabilistic Principal Component Analysis (PCA) (LAWRENCE, 2005).

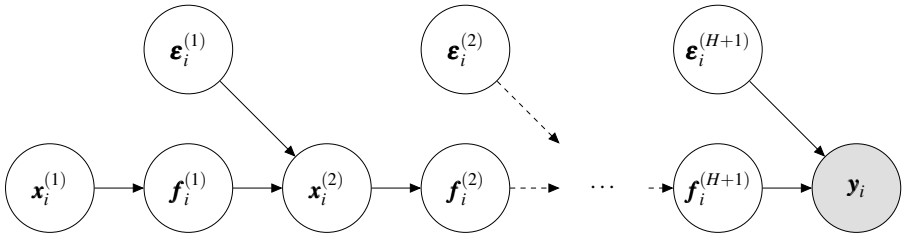


Figure 9 – Graphical model for the Deep GP with $H+1$ layers. Note that if the inputs are made available, the variables $\boldsymbol{x}_i^{(1)}$ in the first layer are not latent anymore.

covariance function with ARD hyperparameters. Even if the variational algorithm increases the model complexity and overall computational requirements, most of the recent models derived from the GP-LVM follow the latter Bayesian approach, including the ones proposed in the next chapters of the present thesis.

2.8 Hierarchical and Deep Gaussian Processes

Since the prior distribution on the latent variables of the GP-LVM framework is a design choice, one could increase the model hierarchy by considering another GP prior for it. Such stack of GP-LVMs was firstly proposed by Lawrence and Moore (2007), who anticipated its application to dynamical modeling and hierarchical data visualization.

The Deep GP framework defined by Damianou and Lawrence (2013) consolidated the concept of hierarchical GP modeling, where the prior in the inputs of a layer is given by a GP modeled by the previous layer. Fig. 9 illustrates the general Deep GP graphical model, which, considering $H+1$ layers, can be described by the equations below:

$$\boldsymbol{y}_i = f^{(H+1)}\left(\boldsymbol{x}_i^{(H+1)}\right) + \boldsymbol{\varepsilon}_i^{(H+1)}, \tag{2.35}$$

$$\boldsymbol{x}_i^{(h+1)} = f^{(h)}\left(\boldsymbol{x}_i^{(h)}\right) + \boldsymbol{\varepsilon}_i^{(h)}, \qquad 1 \leq h \leq H, \tag{2.36}$$

$$\text{where } \boldsymbol{f}_{:d}^{(h)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_f^{(h)}\right), \tag{2.37}$$

where $\boldsymbol{\varepsilon}_i^{(h)} \in \mathbb{R}^{D_h}$ is a Gaussian noise associated with the h-layer and the functions $f^{(h)}(\cdot)$ are vector valued, i.e., output vectors. Following the strategy previously used for the GP-LVM, each d dimension outputted by the functions $f^{(h)}(\cdot)$ is modeled by a separate GP prior, though with the same stack of inputs $\boldsymbol{X}^{(h)} \in \mathbb{R}^{N \times D_h}$ for the h-th layer. Since inference in the Deep GP model is intractable, Damianou and Lawrence (2013) follow the variational Bayes strategy behind Bayesian GP-LVM and approximately integrate the latent variables in all layers of the model.

Such broad class of models presents some interesting features. For instance, a Deep GP is actually no longer a GP, since the multi-layered composition of GP priors cannot be explained by a single GP prior (DAMIANOU, 2015). Furthermore, since each GP layer is integrated out (at least approximately), the model performs process composition. In Section 2.5 we mentioned how a GP model is obtained when we integrate all the weights of a 1-hidden layer neural network with infinite width. Deep GPs have similar parallels with multi-layered neural networks (DUVENAUD *et al.*, 2014).

Besides the original variational approach detailed by Damianou and Lawrence (2013), many other solutions have more recently been proposed as alternatives to perform inference with Deep GPs, such as a nested variational method (HENSMAN; LAWRENCE, 2014), an auto-encoded set-up (DAI *et al.*, 2016), applying approximate Expectation Propagation (BUI *et al.*, 2016), using random Fourier features (CUTAJAR *et al.*, 2016), sampling techniques (WANG *et al.*, 2016) and doubly stochastic variational inference (SALIMBENI; DEISENROTH, 2017). Those approaches present pros and cons, which makes this topic a promising area of research.

Deep GPs generalize other complex GP-based models such as warped GPs, which provide nonlinear transformation of the output space (SNELSON *et al.*, 2004; LÁZARO-GREDILLA, 2012), manifold learning by transformation of the input space (CALANDRA *et al.*, 2016) and deep kernel learning (WILSON *et al.*, 2016a). Besides, the hierarchical composition of GP priors alleviates the problem of choosing task-specific covariance functions, since the successive nonlinear mappings, for instance from the use of the exponentiated quadratic kernel, can be made more expressive than single mappings from shallow models.

Remark In the past few years, numerous successful applications have been presented in the literature where automatic feature learning is performed by deep neural networks, i.e., large parametric models with several layers, usually trained with the backpropagation algorithm, a field which has been called *Deep Learning* (LECUN *et al.*, 2015; SCHMIDHUBER, 2015; GOODFELLOW *et al.*, 2016). Models based on Deep GPs are related nonparametric efforts which bring a Bayesian treatment to uncertainty, a valuable feature that enables learning from smaller datasets and generates outputs with clear probabilistic interpretation.

2.9 Discussion

In this chapter we have summarized the foundations of the GP modeling framework, with focus on its use on regression tasks. At least two important features were emphasized when compared to other nonlinear methods: the clear probabilistic interpretation of the predictions, instead of point estimates, and the comprehensive Bayesian methodology to perform model selection.

We derived the GP predictive expressions from both the prior over the function space view, using the multivariate Gaussian distribution's properties and tractable integrals, and the feature space approach, where we begin from a Bayesian linear regression setting and then are able to analytically marginalize the model parameters.

We have also described some important frameworks that enhance the standard GP approach: the variational sparse GP, for handling larger datasets; the GP-LVM, to perform Bayesian unsupervised learning and training from noisy inputs; and Deep GPs, that enable general hierarchical modeling.

In the next chapters we will use the tools described so far to present some clever GP modeling approaches introduced by several authors and also propose some new methods inspired by such contributions.

3 DYNAMICAL MODELING AND RECURRENT GAUSSIAN PROCESSES MODELS

"Truth is much too complicated
to allow anything but approximations."
(John von Neumann)

The useful features and high applicability of GPs have attracted the attention of the system identification and time series analysis community. In such context, many contributions have been recently proposed to exploit the Bayesian nature of GP and tackle different modeling situations.

In this chapter we briefly review the current literature on dynamical modeling with GPs, highlighting the differences between the main approaches. This initial part follows the extensive review made in the recent theses by McHutchon (2014), Damianou (2015), Frigola-Alcade (2015) and the book by Kocijan (2016). Afterwards, we propose a powerful new model named Recurrent Gaussian Processes (RGP), which incorporates a deep recurrent structure built with the focus on learning dynamics from noisy data and performing free simulation. Furthermore, in order to do inference with the RGP model, we introduce a modified variational framework called Recurrent Variational Bayes (REVARB). We conclude the chapter by presenting several computational experiments to evaluate the new RGP/REVARB approach.

3.1 Dynamical Modeling with GPs

There are many approaches to the task of dynamical modeling with GPs, each strategy being related to different variants proposed by several authors. Thus, as mentioned by Frigola-Alcade (2015), we could refer to a "zoo" of dynamical GP models. In this thesis we are especially interested in nonlinear modeling, but we refer the readers to the comprehensive survey by Pillonetto *et al.* (2014) on kernel methods (including GPs) to linear system identification.

We broadly organize the main contributions found in the literature in two groups: models with *external dynamics* and models with *internal dynamics*. Such presentation is loosely inspired by the nonlinear dynamical modeling taxonomy described by Nelles (2013).

Table 1 – List of regressors used by common model structures with external dynamics.

	$\bar{\boldsymbol{y}}_{i-1}$	$\bar{\boldsymbol{u}}_{i-1}$	$\bar{\boldsymbol{e}}_{i-1}$	$\bar{\boldsymbol{p}}_{i-1}$
NAR	✓			
NFIR		✓		
NARX	✓	✓		
NARMAX	✓	✓	✓	
NOE		✓		✓

3.1.1 GP Models with External Dynamics

The class of models with external dynamics is related to the following generic structure:

$$y_i = f(\boldsymbol{\varphi}_i) + \varepsilon_i^{(y)}, \tag{3.1}$$

where y_i is the output, $\varepsilon_i^{(y)}$ is the observation noise, $f(\cdot)$ is some unknown function and $\boldsymbol{\varphi}_i$ is the so-called *regressor vector*, which contains a combination of measured values, such as external inputs, past observed outputs and predicted errors. Usually such values are given by delayed measures, e.g., for 2 delayed outputs we would have $\boldsymbol{\varphi}_i = [y_{i-1}, y_{i-2}]^\top$. In general, we can write

$$\begin{aligned}
y_i &= f([\bar{\boldsymbol{y}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}, \bar{\boldsymbol{e}}_{i-1}, \bar{\boldsymbol{p}}_{i-1}]) + \varepsilon_i^{(y)}, \qquad (3.2)\\
\bar{\boldsymbol{y}}_{i-1} &= [y_{i-1}, y_{i-2}, \cdots, y_{i-L_y}]^\top,\\
\bar{\boldsymbol{u}}_{i-1} &= [u_{i-1}, u_{i-2}, \cdots, u_{i-L_u}]^\top,\\
\bar{\boldsymbol{e}}_{i-1} &= [e_{i-1}, e_{i-2}, \cdots, e_{i-L_e}]^\top, \quad \text{where } e_i = y_i - \hat{y}_i,\\
\bar{\boldsymbol{p}}_{i-1} &= [\hat{y}_{i-1}, \hat{y}_{i-2}, \cdots, \hat{y}_{i-L_p}]^\top.
\end{aligned}$$

In the previous expressions, the constants L_y, L_u, L_e, L_p are respectively the lagged orders chosen for the outputs y_i, inputs u_i, prediction errors e_i and past predictions $\hat{y}_i$. By removing some of those regressors, we get the common nonlinear structures listed below, whose components are summarized in Tab. 1:

NAR (Nonlinear Autoregressive): uses only past outputs.

NFIR (Nonlinear Finite Impulse response): uses only past inputs.

NARX (Nonlinear Autoregressive with eXogenous inputs): uses past outputs and inputs.

NARMAX (Nonlinear Autoregressive Moving Average with eXogenous inputs): uses past outputs, inputs and prediction errors.

NOE (Nonlinear Output Error): uses past inputs and past predictions.

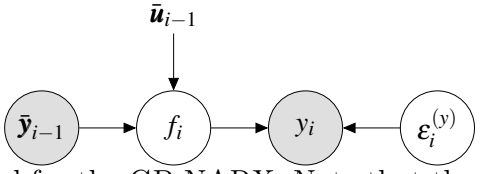


Figure 10 – Graphical model for the GP-NARX. Note that the model input $\boldsymbol{x}_i$ (not shown) is comprised of the regressor vectors $\bar{\boldsymbol{u}}_{i-1}$ and $\bar{\boldsymbol{y}}_{i-1}$, where the former is deterministic and the latter is random, although also treated as deterministic in the more common approaches.

The GP framework can be directly incorporated, for instance, to traditional NARX (or NAR if $L_u = 0$) modeling by considering a Gaussian observation noise and defining regressors for the model input following Tab. 1, which gives rise to the GP-NARX model:

$$y_i = f(\boldsymbol{x}_i) + \varepsilon_i^{(y)}, \quad \varepsilon_i^{(y)} \sim \mathcal{N}(0, \sigma_y^2), \tag{3.3}$$

$$\boldsymbol{x}_i = [\bar{\boldsymbol{y}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}]^\top = [[y_{i-1}, y_{i-2}, \cdots, y_{i-L_y}], [u_{i-1}, u_{i-2}, \cdots, u_{i-L_u}]]^\top, \tag{3.4}$$

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_f), \quad [K_f]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{3.5}$$

where $y_i \in \mathbb{R}$, $\boldsymbol{f} \in \mathbb{R}^N$, σ_y^2 is the noise variance and the model input $\boldsymbol{x}_i \in \mathbb{R}^D$ has dimension $D = L_y + L_u$. By using an ARD covariance function $k(\cdot,\cdot)$, it is possible to automatically select which regressors within the chosen orders range are relevant to the problem. Hyperparameter learning and prediction follow the standard GP regression methodology presented in Chapter 2.

The structure of the GP-NARX model is illustrated in Fig. 10. It is important to note that, although the input of the GP-NARX contains the random vector $\bar{\boldsymbol{y}}_{i-1}$, the more conventional approaches consider it to be deterministic, similar to the exogenous input $\bar{\boldsymbol{u}}_{i-1}$. Thus, the observation noise, which is the only modeled noise, should not be independent, i.e., future outputs are actually conditioned on past noisy outputs. Such unrealistic independent noise assumption is a well known limitation of NARX models in general (NELLES, 2013).

Due to its simplicity and applicability, GP-NARX models are among the most common approaches to dynamical modeling with GPs. Since early work by Murray-Smith *et al.* (1999), Gregorcic and Lightbody (2002), Solak *et al.* (2003), Kocijan *et al.* (2005), many authors have followed this direction, usually tackling specific scenarios or proposing additional features such as nonstationary covariance function for time series prediction (BRAHIM-BELHOUARI; BERMAK, 2004) and nonstationary system identifi-

cation (ROTTMANN; BURGARD, 2010), local modeling (AŽMAN; KOCIJAN, 2011), integrated pre-processing filter (FRIGOLA-ALCADE; RASMUSSEN, 2013), higher-order frequency response functions (WORDEN *et al.*, 2014) and system identification in the presence of outliers (MATTOS *et al.*, 2015).

Girard *et al.* (2003) aimed to overcome the inherent noise inconsistencies of the NARX structure by handling the uncertainty in the regressors caused by the feedback of random variables during multi-step ahead prediction. They approximately propagate the uncertainty using Gaussian approximations and moment matching, but only during the prediction step. Such limitation was handled by McHutchon and Rasmussen (2011), who apply local linear expansions to train GP regression models with noisy inputs. Later, Damianou and Lawrence (2015) considered a variational approximation in both training and test steps, while Bijl *et al.* (2016) approached the task of learning with input noise in an online context.

Although less common than NARX models, there are some efforts in the literature towards GP-based solutions with other external dynamics structures, such as GPs with ARMA noise model (MURRAY-SMITH; GIRARD, 2001), GP-NFIR model applications (ACKERMANN *et al.*, 2011) and GP training with output error (OE) (KOCIJAN; PETELIN, 2011).

3.1.2 GP Models with Internal Dynamics

Models with internal dynamics extend the structures described so far by presenting some form of internal memory. A general formulation can be written in a state-space model (SSM) representation:

$$\boldsymbol{x}_i = f(\boldsymbol{x}_{i-1}, u_{i-1}) + \boldsymbol{\varepsilon}_i^{(x)}, \tag{3.6}$$

$$y_i = g(\boldsymbol{x}_i) + \varepsilon_i^{(y)}, \tag{3.7}$$

where $\boldsymbol{\varepsilon}_i^{(x)} \in \mathbb{R}^D$ is the *transition* (or *process*) noise, $f(\cdot)$ is now called the *transition* function, $g(\cdot)$ is the *observation* (or *emission* function) and the vector $\boldsymbol{x}_i \in \mathbb{R}^D$ is called the *state* of the system related to the instant i. The main differences when compared to the previous models with external dynamics are: **i)** the inclusion of a separate noise model for the dynamics; **ii)** the existence of separate functions to model the transition and the observation; **iii)** the definition of the state $\boldsymbol{x}_i$, which acts as an internal memory

with general structure.

From such characterization, we can see that SSMs do not use lagged versions of the outputs to learn dynamics and hence do not pose much prior restrictions to the states. However, it is important to emphasize that in most applications the states $\boldsymbol{x}_i$ are *latent*, i.e., not observed directly.

SSMs usually involve three important operations, which from the Bayesian point of view are related to three conditional distributions, listed below (SÄRKKÄ, 2013):

Filtering Estimation of a state from the observed outputs up until the current instant, i.e., the posterior distribution $p(\boldsymbol{x}_i|y_i, y_{i-1}, \cdots, y_1)$;

Prediction Estimation of a state given only previous observations, i.e., the conditioning distribution $p(\boldsymbol{x}_i|y_{i-1}, y_{i-2}, \cdots, y_1)$;

Smoothing Estimation of the states after the observation of all available measured output, i.e., the posterior distribution $p(\boldsymbol{x}_i|y_N, y_{N-1}, \cdots, y_1)$.

Certainly, the above distributions can also be conditioned to external inputs $u_i|_{i=1}^N$, if they are available.

In a GP-SSM context we have the GP priors below:

$$\boldsymbol{F} \sim \prod_{d=1}^{D} \mathcal{N}(\boldsymbol{f}_{:d}|\boldsymbol{0}, \boldsymbol{K}_f), \quad \text{where } \boldsymbol{f}_{:d} = [\boldsymbol{f}_i]_d|_{i=1}^N = F_{id}|_{i=1}^N, \tag{3.8}$$

$$\boldsymbol{g} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_g), \tag{3.9}$$

where $\boldsymbol{g} \in \mathbb{R}^N$ is the vector of latent outputs from $g(\cdot)$ and $\boldsymbol{F} \in \mathbb{R}^{N \times D}$ is the collection of D latent transition vectors $\boldsymbol{f}_{:d} \in \mathbb{R}^N$, $1 \leq d \leq D$. Note that each output dimension d of $f(\cdot)$, associated with each state dimension, is modeled by a separate GP, although with the same inputs.

The graphical model for the GP-SSM, illustrated in Fig. 11, highlights the challenges faced by this model, since the latent states $\boldsymbol{x}_i$ and latent variables $\boldsymbol{f}_i \in \mathbb{R}^D$ and g_i, related respectively to the transition and observation functions, must be marginalized somehow. Such latent structure and the recurrent nature of $f(\cdot)$ bring several intractabilities to the model, so regular GP expressions are not analytical anymore and approximate methods become necessary.

It is interesting to note that in the GP-NARX model shown in Fig. 10 the regressors used as inputs, i.e., $\boldsymbol{x}_i = [\bar{\boldsymbol{y}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}]^\top$, can actually be seen as an observable state

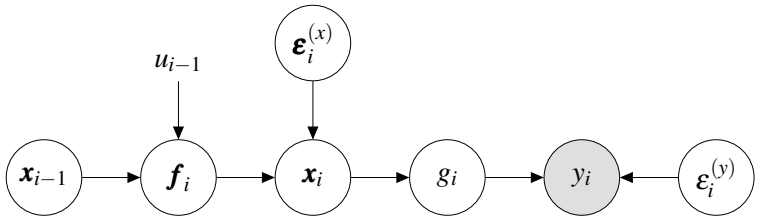


Figure 11 – Graphical model for the GP-SSM. Besides the deterministic external input u_{i-1}, the only observed variable is the noisy observation y_i. Note that both the transition and observation functions, related respectively to $\boldsymbol{f}_i$ and g_i, have random variables as inputs.

in a SSM context. Conversely, the GP-SSM latent state can be roughly seen as a vector of unobserved regressors.

Many authors have explored different approaches to perform inference with the general GP-SSM framework, such as the use of expectation maximization (EM) (TURNER *et al.*, 2010), particle Markov Chain Monte Carlo (PMCMC) methods (FRIGOLA-ALCADE *et al.*, 2013; SVENSSON *et al.*, 2016) and hybridization of variational Bayes and sequential Monte Carlo (SMC) (FRIGOLA-ALCADE *et al.*, 2014).

The works mentioned above aim to perform Bayesian filtering or smoothing, i.e., finding posterior distribution to the state $\boldsymbol{x}_i$ given observations up to $\boldsymbol{y}_i$ or all the outputs $\boldsymbol{y}_i|_{i=1}^N$, respectively. In the linear case with Gaussian noise, Kalman filtering (KALMAN *et al.*, 1960) and Kalman smoothing (RAUCH *et al.*, 1965) provide the optimal solutions. Although several nonlinear extensions exist (and are comprehensively presented by Särkkä (2013)), the previously cited works constitute nonparametric GP-based alternatives. We refer the readers to the work by Reece and Roberts (2010) and Särkkä *et al.* (2013) for detailed connections between GP models and Kalman filtering/smoothing methods.

Other authors have opted to apply variants of the GP-LVM framework (LAWRENCE, 2004; TITSIAS; LAWRENCE, 2010) to propose alternative GP models with internal dynamics (WANG *et al.*, 2005; LAWRENCE; MOORE, 2007; FERRIS *et al.*, 2007; DAMIANOU *et al.*, 2011). As opposed to sampling methods, those approaches usually follow deterministic approximations, such as finding a *maximum a posteriori* (MAP) solution for the latent states or approximate marginalization via variational algorithms.

Each aforementioned work usually considers slightly different model formulations, aiming to simplify the inference step or to cover task specific requirements. For instance, the Gaussian Process Dynamical Model (GPDM) proposed by Wang *et al.* (2005) consists of starting from the GP-LVM framework, maintaining the original GP prior $p(\boldsymbol{Y}|\boldsymbol{X})$

between the latent (state) space and the observations, and then includes a transition distribution $p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})$ between states. Ferris *et al.* (2007) follow a similar approach, but instead of directly modeling such transition distribution, the prior on the latent space $p(\boldsymbol{X})$ is modified by imposing positioning constraints between successive points. Similar to the original GP-LVM (LAWRENCE, 2005), in both cases the authors proceed by maximizing the latent variables $\boldsymbol{x}_i$ along with the model hyperparameters, i.e., a MAP solution for the states. This implies treating the latent variables $\boldsymbol{x}_i$ as model parameters themselves, since they are not marginalized.

Damianou *et al.* (2011) proposed the Variational Gaussian Process Dynamical Systems (VGPDS), which approximately integrates the latent variables in its structure using the variational approach of the Bayesian GP-LVM framework, introduced by Titsias and Lawrence (2010). However, differently from the GPDM, the transition function $f(\cdot)$ in the expression $\boldsymbol{x}_i = f(\boldsymbol{x}_{i-1}) + \boldsymbol{\varepsilon}_i^{(x)}$ is modeled by a GP with the time instants $\boldsymbol{t}$ as inputs, i.e., $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_f), [K_f]_{ij} = k(t_i, t_j)$. It is argued by Frigola-Alcade (2015) that such model is capable of handling only linear dynamics, since the transition between states is not modeled directly, but implicitly by depending from the same time inputs. Still, Damianou *et al.* (2011) demonstrated how VGPDS can be easily applied to high-dimensional dynamical data, such as video sequences.

Alvarez *et al.* (2009) pursue a very unique approach, named *latent force models*, a hybrid GP model which applies linear differential equations in its covariance function in order to incorporate physical knowledge about the underlying dynamical system. Later, Alvarez *et al.* (2010) extended that technique to enable switching between different dynamics, allowing its use in the task of nonlinear robot movement representation.

Finally, it is worth noting the trend in combining GP-based dynamical models with other powerful learning methods, such as neural networks. For instance, Chatzis and Demiris (2011) builds a GP model enhanced with the *reservoir computing* approach from echo state networks (ESN) (JAEGER, 2001) to obtain an echo state GP (ESGP) for dynamical data modeling. More recently, Al-Shedivat *et al.* (2016) propose a dynamical GP model where the covariance function is replaced by a long short-term memory (LSTM) recurrent neural network (HOCHREITER; SCHMIDHUBER, 1997). The resulting GP-LSTM model aims to retain the nonparametric probabilistic nature of GPs while allowing direct learning of the recurrent kernel via the LSTM.

3.2 Recurrent GPs

Though simple and tractable, the GP-NARX model results in inconsistent noise assumptions and mishandles the uncertainty in its basic formulation. The GP-SSM admits unobserved variables, distinct noises and separately models the transition and observation functions. However, its definition of the latent states is vague and ambiguous, since usually there is no strong *a priori* assumptions about them. Besides, approximate methods are necessary, increasing the overall inference complexity when compared to GP-NARX.

With those observations in mind, we propose an alternative SSM approach where the states have an specific autoregressive structure. Differently from standard NARX models, the autoregression in our model is performed with latent variables modeled by probability distributions. Such structure is defined by the set of equations below:

$$x_i = f(\bar{\boldsymbol{x}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}) + \varepsilon_i^{(x)}, \tag{3.10}$$

$$\boldsymbol{y}_i = g(\bar{\boldsymbol{x}}_i) + \boldsymbol{\varepsilon}_i^{(y)}, \tag{3.11}$$

where following the previously defined notation for regressors and given L lag steps we have $\bar{\boldsymbol{x}}_{i-1} = [x_{i-1}, \cdots, x_{i-L}]^\top$ and $\bar{\boldsymbol{u}}_{i-1} = [u_{i-1}, \cdots, u_{i-L_u}]^\top$. Even if the output of the transition function $f(\cdot)$ in Eq. (3.10) is chosen to be 1-dimensional, it should be noticed that the actual hidden state $\bar{\boldsymbol{x}}_i \in \mathbb{R}^L$ is multidimensional for $L > 1$. Since such latent transition variables are also used as inputs in the next iterations, our model can be characterized as *recurrent*.

As argued by Pascanu *et al.* (2014) in the context of recurrent neural networks (RNNs), a hierarchical (or *deep*) structure can be helpful for recurrent modeling. Besides, such multilayer structure has been successfully exploited more recently with the rise of deep GP models (DAMIANOU; LAWRENCE, 2013; HENSMAN; LAWRENCE, 2014; DAMIANOU, 2015), as briefly described in Section 2.8.

If we consider H transition functions, each one comprising a hidden layer, it naturally results in the following deep recurrent structure:

$$x_i^{(h)} = f^{(h)}\left(\hat{\boldsymbol{x}}_i^{(h)}\right) + \varepsilon_i^{(h)}, \qquad \boldsymbol{f}^{(h)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_f^{(h)}\right), \qquad 1 \leq h \leq H, \tag{3.12}$$

$$\boldsymbol{y}_i = f^{(H+1)}\left(\hat{\boldsymbol{x}}_i^{(H+1)}\right) + \boldsymbol{\varepsilon}_i^{(H+1)}, \qquad \boldsymbol{f}^{(H+1)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_f^{(H+1)}\right) \tag{3.13}$$

where we have put GP priors with zero mean and covariance matrix $\boldsymbol{K}_f^{(h)}$ on the unknown functions $f(\cdot)^{(h)}$, the noise in each layer is defined as $\boldsymbol{\varepsilon}_i^{(h)} \sim \mathcal{N}(0, \sigma_h^2)$ and the upper index

differentiates variables and functions from distinct layers. We also introduce the notation

$$\hat{\boldsymbol{x}}_i^{(h)} = \begin{cases} \left[\bar{\boldsymbol{x}}_{i-1}^{(1)}, \bar{\boldsymbol{u}}_{i-1}\right]^\top = \left[\left[x_{i-1}^{(1)}, \cdots, x_{i-L}^{(1)}\right], \left[u_{i-1}, \cdots, u_{i-L_u}\right]\right]^\top, & \text{if } h = 1, \\ \left[\bar{\boldsymbol{x}}_{i-1}^{(h)}, \bar{\boldsymbol{x}}_i^{(h-1)}\right]^\top = \left[\left[x_{i-1}^{(h)}, \cdots, x_{i-L}^{(h)}\right], \left[x_i^{(h-1)}, \cdots, x_{i-L+1}^{(h-1)}\right]\right]^\top, & \text{if } 1 < h \leq H, \\ \bar{\boldsymbol{x}}_i^{(H)} = \left[x_i^{(H)}, \cdots, x_{i-L+1}^{(H)}\right]^\top, & \text{if } h = H+1. \end{cases} \tag{3.14}$$

We note that the vector $\bar{\boldsymbol{x}}_i^{(h)}$ represents the autoregressive state associated with the layer h in the instant i, from which the dynamics are learned[1]. Since the vectors $\bar{\boldsymbol{x}}_i^{(h)}$ are not directly dependent on past observations, they are related to general latent states of regular SSMs. Thus, in the nonlinear system identification terminology that we have presented in this chapter, inspired by Nelles (2013), the RGP is considered a model with *internal* dynamics.

The RGP model can also be written using only the involved distributions:

$$p\left(\boldsymbol{f}^{(h)} \middle| \hat{\boldsymbol{X}}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{f}^{(h)} \middle| \boldsymbol{0}, \boldsymbol{K}_f^{(h)}\right), \qquad 1 \leq h \leq H+1, \tag{3.15}$$

$$p\left(x_i^{(h)}\right) = \mathcal{N}\left(x_i^{(h)} \middle| \mu_{0i}^{(h)}, \lambda_{0i}^{(h)}\right), \qquad 1 \leq i \leq L, \tag{3.16}$$

$$p\left(x_i^{(h)} \middle| f_i^{(h)}\right) = \mathcal{N}\left(x_i^{(h)} \middle| f_i^{(h)}, \sigma_h^2\right), \qquad L+1 \leq i \leq N, \tag{3.17}$$

$$p\left(y_i \middle| f_i^{(H+1)}, \sigma_{H+1}^2\right) = \mathcal{N}\left(y_i \middle| f_i^{(H+1)}, \sigma_{H+1}^2\right), \qquad L+1 \leq i \leq N, \tag{3.18}$$

where $\hat{\boldsymbol{X}}^{(h)}$ is simply given by stacking the vectors $\hat{\boldsymbol{x}}_i^{(h)}|_{i=L+1}^N$ and we have made explicit the means $\mu_{0i}^{(h)}$ and variances $\lambda_{0i}^{(h)}$ of the Gaussian priors in the initial L latent variables of each layer[2]. The former set of equations results in a full probabilistic model, where each model component is given a distribution and from which we could sequentially take samples.

The graphical model for the RGP is presented in Fig. 12, where we have kept the state notation $\bar{\boldsymbol{x}}^{(h)}$ to make the recurrent dependencies more clear. Fig. 13 details the recurrent connections in a single transition layer. It should be noted that the standard GP-NARX and GP-SSM can also be seen as RGPs, but with different states structure.

We emphasize that the RGP model preserves the unobserved states of standard SSMs but avoids the ambiguities of generic multidimensional states by imposing a specific

[1] We emphasize the different notations presented in Eq. (3.14): $x_i^{(h)}$ is the dynamical latent variable, $\bar{\boldsymbol{x}}_{i-1}^{(1)}$ is the latent state vector and $\hat{\boldsymbol{x}}_i^{(h)}$ is the h-th layer input.

[2] Note that in Eq. (3.18) we write the index range of the observations y_i from $i = L+1$ to $i = N$, comprising $N-L$ elements, since the first L elements are considered as initial conditions. The same strategy is applied to the latent variables in the hidden layers in Eq. (3.17).

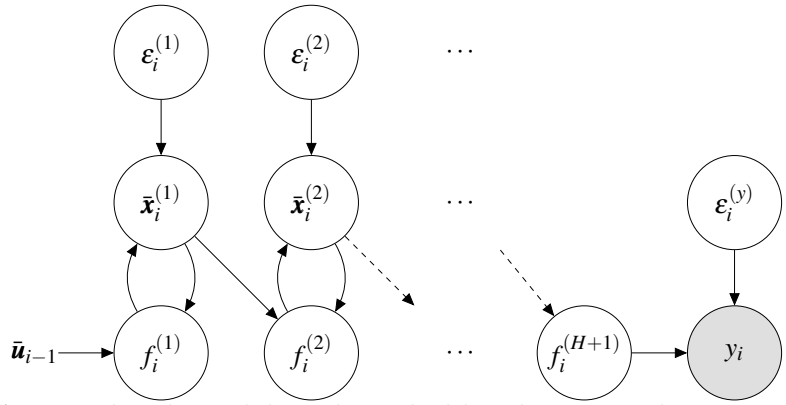


Figure 12 – RGP graphical model with H hidden layers. The recurrent hierarchical structure illustrates how the latent variables outputted by a given transition layer are used as inputs in that layer and in the next one. Similar to SSMs, the only observed variables are the deterministic external inputs and the noisy measurements.

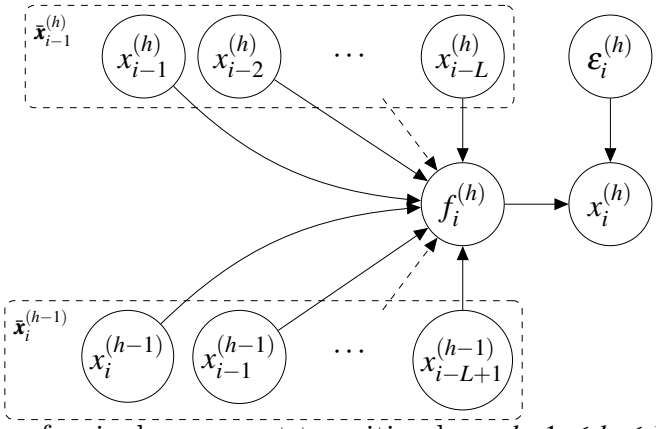


Figure 13 – Detailing of a single recurrent transition layer h, $1 \leq h \leq H$, of the RGP model. Note that for $h = 1$, the variables $x_{i-l+1}^{(h-1)}|_{l=1}^{L}$ are replaced by the deterministic external inputs $u_{i-l}|_{l=1}^{L_u}$. If these are not available, they are simply omitted.

latent autoregressive structure. It is also worth mentioning that our RGP model, as defined by Eqs. (3.12) and (3.13), can be seen as a special case of the general Deep GP framework (briefly described in Section 2.8) where the priors of the latent variables in each hidden layer follow the autoregressive structure of Eq. (3.14), illustrated in Fig. 12. Thus, the RGP model inherits the powerful expressiveness and uncertainty handling properties of Deep GPs, as well as their intractabilities.

So far we have only introduced the structure and the probabilistic expressions that describe the RGP model, without worrying about how to use it to learn dynamics from data or perform predictions. This agrees with our goal of separating model from inference algorithm, as argued in Section 1.3. In the next section we present the REVARB framework, a variational method to perform inference with this novel RGP model, which

is able to handle the uncertainties from both the data and the model components.

Remark A given model is called *generative* if it defines a joint distribution over inputs and outputs, which enables it to generate data (KOLLER; FRIEDMAN, 2009). The RGP model considers distributions in Eqs. (3.15)-(3.18) for all its components but the exogenous inputs $\boldsymbol{u}_i|_{i=1}^N$, which are actually absent if a time series is being modeled. If such inputs are present, we could easily include some Gaussian distribution for them and treat them similar to the latent dynamical variables $x_i^{(1)}|_{i=1}^N$ of the first layer. We comment about that strategy within the REVARB framework in Appendix A.1. Thus, we can safely refer to RGPs as generative models.

3.3 REVARB: REcurrent VARiational Bayes

Inference is intractable in our RGP model because we are not able to get analytical forms for the posterior of the latent function values $\boldsymbol{f}^{(h)}$ or the marginal likelihood of the observations $\boldsymbol{y}$, as opposed to what we have done for standard GP regression in Sections 2.3 and 2.4. We tackle such intractabilities with the variational approximation scheme named REVARB: REcurrent VARiational Bayes.

Thus, we have two goals in this section: (i) approximately marginalize all the latent variables of the RGP model, (ii) find an expression that approximates the RGP model evidence, i.e., the marginal log-likelihood, which can be used to perform model selection. The REVARB framework, described henceforward, covers both goals.

REVARB is based on the variational sparse framework proposed by Titsias (2009a) and described in Section 2.6.1. Thus, we start by including to each layer h a number of M inducing points $\boldsymbol{z}^{(h)} \in \mathbb{R}^M$ evaluated in M pseudo-inputs $\boldsymbol{\zeta}_j^{(h)}|_{j=1}^M \in \mathbb{R}^{D_h}$, where D_h is the same dimension of the layer h input $\hat{\boldsymbol{x}}_i^{(h)}$. Such inducing points $\boldsymbol{z}^{(h)}$ are extra samples of the GP that models $f^{(h)}(\cdot)$ and we can write $p\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{0}, \boldsymbol{K}_z^{(h)}\right)$, where $\boldsymbol{K}_z^{(h)} \in \mathbb{R}^{M \times M}$ is the covariance matrix obtained from $\boldsymbol{\zeta}_j^{(h)}|_{j=1}^M$.

From the expressions defined in past section, the joint distribution relating all

the variables of the augmented model is now given by:

$$p\left(\boldsymbol{y}, \left\{\boldsymbol{x}^{(h)}\right\}\Big|_{h=1}^{H}, \left\{\boldsymbol{f}^{(h)}, \boldsymbol{z}^{(h)}\right\}\Big|_{h=1}^{H+1}\right) = \\ \left(\prod_{i=L+1}^{N} p\left(y_i \middle| f_i^{(H+1)}\right) p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H)}\right) \prod_{h=1}^{H} p\left(x_i^{(h)} \middle| f_i^{(h)}\right) p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right)\right) \\ \left(\prod_{h=1}^{H+1} p\left(\boldsymbol{z}^{(h)}\right)\right) \left(\prod_{i=1}^{L}\prod_{h=1}^{H} p\left(x_i^{(h)}\right)\right), \tag{3.19}$$

where the boldface indexless notation $\boldsymbol{x}^{(h)}$ is used to refer to all variables $x_i^{(h)}, \forall i \in \{1, \cdots, N\}$, in a given layer h. Note that we have omitted for now the dependence on the pseudo-inputs $\boldsymbol{\zeta}_j^{(h)}$. It is important to emphasize that such augmentation does not fundamentally change the original model, since if we integrate out the inducing points $\boldsymbol{z}^{(h)}$ we recover the same original model expressions.

The exact marginal likelihood $p(\boldsymbol{y})$ still cannot be computed analytically. For instance, in order to compute it we would need to integrate all the latent variables in Eq. (3.19). However, such integration is not analytical, since the latent variables $x_i^{(h)}$ appear in the terms $\left|\boldsymbol{K}_f^{(h)}\right|$ and $\left(\boldsymbol{K}_f^{(h)}\right)^{-1}$ inside the GP priors on $\boldsymbol{f}^{(h)}$.

Fortunately, we are able to lower bound the marginal log-likelihood $\log p(\boldsymbol{y})$ by applying Jensen's inequality, similar to the standard variational approach (also pursued by the sparse approximation described in Section 2.6.1):

$$\log p(\boldsymbol{y}) \geq \int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} Q \log \left[\frac{p\left(\boldsymbol{y}, \left\{\boldsymbol{x}^{(h)}\right\}\Big|_{h=1}^{H}, \left\{\boldsymbol{f}^{(h)}, \boldsymbol{z}^{(h)}\right\}\Big|_{h=1}^{H+1}\right)}{Q}\right], \tag{3.20}$$

where Q is the variational distribution. We note that, as demonstrated in Section 2.6.1 in the context of the variational sparse framework, the maximization of the bound in Eq. (3.20) is equivalent to minimizing the KL divergence between the variational posterior Q and the true posterior. We conveniently choose the following factorized expression for Q:

$$Q = \left(\prod_{h=1}^{H} q\left(\boldsymbol{x}^{(h)}\right)\right) \left(\prod_{h=1}^{H+1} q\left(\boldsymbol{z}^{(h)}\right)\right) \left(\prod_{i=L+1}^{N}\prod_{h=1}^{H+1} p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right)\right). \tag{3.21}$$

In the former equation, the distribution $p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right)$ can be found by using the Gaussian conditioning property, since $\boldsymbol{f}^{(h)}$ and $\boldsymbol{z}^{(h)}$ come from the same GP:

$$p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right) = \mathcal{N}\left(f_i^{(h)} \middle| \left[\boldsymbol{a}_f^{(h)}\right]_i, \left[\boldsymbol{\Sigma}_f^{(h)}\right]_{ii}\right), \tag{3.22}$$

$$\text{where } \boldsymbol{a}_f^{(h)} = \boldsymbol{K}_{fz}^{(h)} \left(\boldsymbol{K}_z^{(h)}\right)^{-1} \boldsymbol{z}^{(h)},$$

$$\text{and } \boldsymbol{\Sigma}_f^{(h)} = \boldsymbol{K}_f^{(h)} - \boldsymbol{K}_{fz}^{(h)} \left(\boldsymbol{K}_z^{(h)}\right)^{-1} \left(\boldsymbol{K}_{fz}^{(h)}\right)^{\top}.$$

In the above expression, $\boldsymbol{K}_f^{(h)} \in \mathbb{R}^{(N-L)\times(N-L)}$ is the standard covariance matrix obtained from $\hat{\boldsymbol{x}}^{(h)}$, $\boldsymbol{K}_z^{(h)} \in \mathbb{R}^{M\times M}$ is the covariance matrix calculated from the pseudo-inputs $\boldsymbol{\zeta}^{(h)}$ and the components of the matrix $\boldsymbol{K}_{fz}^{(h)} \in \mathbb{R}^{(N-L)\times M}$ are calculated by $\left[\boldsymbol{K}_{fz}^{(h)}\right]_{ij} = k\left(\hat{\boldsymbol{x}}_i^{(h)}, \boldsymbol{\zeta}_j^{(h)}\right)$. We have also used in Eq. (3.22) the subindex $[\cdot]_i$ to indicate the i-th element of the vector $\boldsymbol{a}_f^{(h)}$ and the double subindex $[\cdot]_{ii}$ to indicate the i-th element of the diagonal of the matrix $\boldsymbol{\Sigma}_f^{(h)}$. It should be noted that we do not need to compute the full covariance matrix $\boldsymbol{K}_f^{(h)}$, but only its $N-L$ diagonal elements, a feature inherited from the sparse approximation. Moreover, we recall that the number of inducing points M is usually much lower than the number of samples N.

We also consider a factorized mean-field approximation (PARISI, 1988) for $q\left(\boldsymbol{x}^{(h)}\right)$:

$$q\left(\boldsymbol{x}^{(h)}\right) = \prod_{i=1}^{N} q\left(x_i^{(h)}\right) = \prod_{i=1}^{N} \mathcal{N}\left(x_i^{(h)} \middle| \mu_i^{(h)}, \lambda_i^{(h)}\right), \tag{3.23}$$

where $\mu_i^{(h)} \in \mathbb{R}$ and $\lambda_i^{(h)} \in \mathbb{R}_{>0}$ are variational parameters that characterize the posterior approximations. The variational distribution in Eq. (3.23) indicates that the latent variables $\boldsymbol{x}_i^{(h)}$ in a given hidden layer are related to $2N$ variational parameters. In standard variational GP-SSMs, such as the one presented by Frigola-Alcade *et al.* (2014), we would have a total of $2ND$ parameters, for D-dimensional states, even for a diagonal covariance matrix in the posterior. Such reduction of parameters in our mean-field approximation was enabled by the latent autoregressive structure of the RGP model.

Following Titsias (2009a), it turns out that the variational distribution $q\left(\boldsymbol{z}^{(h)}\right)$ that maximizes the bound is a multivariate Gaussian, which can be generically written as $q\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{m}^{(h)}, \boldsymbol{S}^{(h)}\right)$, where $\boldsymbol{m}^{(h)}$ and $\boldsymbol{S}^{(h)}$ are additional variational parameters respectively related to the mean and covariance matrix of the distribution. Fortunately, those can be found analytically and optimally eliminated from the expressions (TITSIAS, 2009a; TITSIAS; LAWRENCE, 2010). This step is detailed in the Appendix A.2.

Replacing the definition of the joint distribution (Eq. (3.19)) and the factorized variational distribution Q (Eq. (3.21)) in the Jensen's inequality of Eq. (3.20), we are able

to cancel the terms $p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)$ inside the logarithm:

$$\begin{aligned}\log p(\boldsymbol{y}) \geq & \sum_{i=L+1}^{N}\int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) p\left(f_i^{(H+1)}\middle|\boldsymbol{z}^{(H+1)},\hat{\boldsymbol{x}}_i^{(H+1)}\right)\log p\left(y_i\middle|f_i^{(H+1)}\right)\\ & + \sum_{i=L+1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}}\left(\prod_{h'=1}^{H} q\left(\boldsymbol{x}^{(h')}\right)\right) q\left(\boldsymbol{z}^{(h)}\right) p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\log p\left(x_i^{(h)}\middle|f_i^{(h)}\right)\\ & - \sum_{i=1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right)\\ & - \sum_{h=1}^{H+1}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log q\left(\boldsymbol{z}^{(h)}\right) + \sum_{h=1}^{H+1}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log p\left(\boldsymbol{z}^{(h)}\right).\end{aligned} \tag{3.24}$$

The former expression can be rewritten in a more compact way as follows:

$$\log p(\boldsymbol{y}) \geq \sum_{i=L+1}^{N}\sum_{h=1}^{H+1}\mathscr{L}_i^{(h)} + \sum_{i=1}^{N}\sum_{h=1}^{H}\mathscr{H}_i^{(h)} + \sum_{i=1}^{L}\sum_{h=1}^{H}\mathscr{L}_{0i}^{(h)} - \sum_{h=1}^{H+1}\mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right)\middle\|p\left(\boldsymbol{z}^{(h)}\right)\right), \tag{3.25}$$

where we have denoted the terms below:

$$\begin{aligned}\mathscr{L}_i^{(H+1)} &= \left\langle p\left(f_i^{(H+1)}\middle|\boldsymbol{z}^{(H+1)},\hat{\boldsymbol{x}}_i^{(H+1)}\right)\log p\left(y_i\middle|f_i^{(H+1)}\right)\right\rangle_{q(\boldsymbol{x})q(\boldsymbol{z})},\\ \mathscr{L}_i^{(h)} &= \left\langle p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\log p\left(x_i^{(h)}\middle|f_i^{(h)}\right)\right\rangle_{q(\boldsymbol{x})q(\boldsymbol{z})},\ 1\leq h\leq H,\\ \mathscr{H}_i^{(h)} &= -\left\langle\log q\left(x_i^{(h)}\right)\right\rangle_{q(\boldsymbol{x})},\ 1\leq h\leq H,\\ \mathscr{L}_{0i}^{(h)} &= \left\langle\log p\left(x_i^{(h)}\right)\right\rangle_{q(\boldsymbol{x})},\ 1\leq h\leq H,\\ \mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right)\middle\|p\left(\boldsymbol{z}^{(h)}\right)\right) &= \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log q\left(\boldsymbol{z}^{(h)}\right) - \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log p\left(\boldsymbol{z}^{(h)}\right),\ 1\leq h\leq H+1.\end{aligned}$$

In the above expressions, the notation $\langle\cdot\rangle_{p(\cdot)}$ indicates expectation with respect to the distributions in the subindex. We omit some of the layer's indexes in such distributions in order to avoid notation clutter, but those agree with the variables inside the expectation.

It is worth interpreting the components of the bound expressed in Eq. (3.25). First of all, $\mathscr{L}_i^{(H+1)}$ is the only term that contains the observations $\boldsymbol{y}$, being the main *data fitting* component. The terms $\mathscr{L}_i^{(h)}\big|_{h=1}^{H}$, on the other hand, are more directly responsible for learning the *transition dynamics* from the latent inputs $\hat{\boldsymbol{x}}_i$ in each hidden layer. The terms $\mathscr{L}_{0i}^{(h)}\big|_{h=1}^{H}$, $1\leq i\leq L$, are related to the *initial conditions* of the latent dynamical variables and are heavily influenced by the priors $p\left(x_i^{(h)}\right)$. The entropy terms $\mathscr{H}_i^{(h)}\big|_{h=1}^{H}$ and the KL divergences $\mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right)\middle\|p\left(\boldsymbol{z}^{(h)}\right)\right)\big|_{h=1}^{H}$ are responsible for constraining the flexibility of the learned latent space and limiting the complexity of the model, acting as *regularizers* that naturally appear in the derived bound.

Working Eq. (3.25) we obtain the final REVARB expression, whose detailed derivation is presented in the Appendix A.2. The lower bound to the model log-marginal likelihood is given by

$$\begin{aligned}
\log p(\boldsymbol{y}) \geq & -\frac{N-L}{2}\sum_{h=1}^{H+1}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_{H+1}^2}\left(\boldsymbol{y}^\top\boldsymbol{y} + \Psi_0^{(H+1)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right)\right) \\
& + \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right| \\
& + \frac{1}{2(\sigma_{H+1}^2)^2}\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top\boldsymbol{y} \\
& + \sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} + \Psi_0^{(h)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right. \\
& + \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right| \\
& + \frac{1}{2(\sigma_h^2)^2}\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} \\
& \left. - \sum_{i=1}^{N}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right)\right\},
\end{aligned} \tag{3.26}$$

where we have to compute some statistics that come up in the bound after solving the expectations:

$$\begin{aligned}
\Psi_0^{(h)} &= \mathrm{Tr}\left(\left\langle \boldsymbol{K}_f^{(h)}\right\rangle_{q(\cdot)^{(h)}}\right) \\
\boldsymbol{\Psi}_1^{(h)} &= \left\langle \boldsymbol{K}_{fz}^{(h)}\right\rangle_{q(\cdot)^{(h)}} \\
\boldsymbol{\Psi}_2^{(h)} &= \left\langle \left(\boldsymbol{K}_{fz}^{(h)}\right)^\top\boldsymbol{K}_{fz}^{(h)}\right\rangle_{q(\cdot)^{(h)}}
\end{aligned} \quad \Rightarrow q(\cdot)^{(h)} = \begin{cases} q\left(\boldsymbol{x}^{(1)}\right), & \text{if } h = 1, \\ q\left(\boldsymbol{x}^{(h)}\right)q\left(\boldsymbol{x}^{(h-1)}\right), & \text{if } 1 < h \leq H, \\ q\left(\boldsymbol{x}^{(H)}\right), & \text{if } h = H+1, \end{cases} \tag{3.27}$$

where $\langle\cdot\rangle_{q(\boldsymbol{x}^{(h)})}$ means expectation with respect to the distribution $q\left(\boldsymbol{x}^{(h)}\right)$, which itself depends only on the variational parameters $\mu_i^{(h)}$ and $\lambda_i^{(h)}$. All the expectations are tractable for our choice of the exponentiated quadratic covariance function and follow expressions similar to the ones presented by Titsias and Lawrence (2010) for the Bayesian GP-LVM, which are detailed in the Appendix A.1. The additional integrals in the last line of Eq. (3.26) involve only Gaussians and are hence also tractable. Note that in the final bound all the latent variables were, at least approximately, marginalized.

The REVARB lower bound acts as a proxy for the log-marginal likelihood and by making it tighter, i.e., maximizing it, the approximation gets closer to the true

expression for $\log p(\boldsymbol{y})$. The bound can be optimized with the help of analytical gradients with respect to the kernel hyperparameters and variational parameters. Note that those are not model parameters[3], so our approach preserves the original nonparametric nature of GP models.

Any gradient-based optimization algorithm can be used to iteratively maximize the REVARB bound. In this work we use the well known BFGS method (FLETCHER, 2013), more specifically the R implementation of the standard *stats* package (R Core Team, 2017). Note that the moments of the Gaussian priors $p\left(x_i^{(h)}\right) = \mathcal{N}\left(x_i^{(h)} \middle| \mu_{0i}^{(h)}, \lambda_{0i}^{(h)}\right)$ of the initial latent variables $x_i^{(h)}|_{i=1}^{L}$ can be either fixed, e.g., $\mu_{0i}^{(h)} = 0$ and $\lambda_{0i}^{(h)} = 1$, or optimized along with the other components of the model. It is important to notice that the full REVARB bound in Eq. (3.26) is not factorized along the observations, as the compact version in Eq. (3.25) could indicate. Thus, a *batch* optimization must be performed.

Remark The presented REVARB derivation does not explicit the filtering or smoothing steps. Indeed, the posterior distribution of the latent states $\bar{\boldsymbol{x}}_i^{(h)}$ in the hidden layers is not calculated, but directly approximated by the variational distributions $q\left(\boldsymbol{x}^{(h)}\right)$, as shown in Eq. (3.23). Thus, after the optimization of the analytical lower bound to the log-marginal likelihood in Eq. (3.26), the variational distributions act as the posterior of the states' components given the training observations, resulting in a smoothing "by-product".

3.3.1 Making Predictions with the REVARB Framework

The REVARB framework allows for a natural way to approximately propagate the uncertainty during both training and testing. Since the input of each layer in a given iteration is a distribution that models the uncertainty in that moment, the computation of the exact predictive distribution is intractable, in opposition to standard GP regression. However, we can still calculate its moments and approximate it by a Gaussian, which can be used in further predictions. For this purpose, we follow the methodology presented in Girard *et al.* (2002), Girard *et al.* (2003) for multi-step ahead GP predictions.

Given a new input $\hat{\boldsymbol{x}}_*^{(h)}$ in the h-th layer, we want to approximate the predictive

[3]As argued in Section 2.2, the kernel hyperparameters are responsible for characterizing the chosen covariance function and the variational parameters (pseudo-inputs and moments of the variational distributions) are related to the inference method.

distribution of the associated output $f_*^{(h)}$:

$$p\left(f_*^{(h)}\right) = \left\langle p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right)\right\rangle_{q(\boldsymbol{x}_*)}, \tag{3.28}$$

where $q(\boldsymbol{x}_*)$ (with omitted layer index) is the approximate distribution of the latent variables in the input $\hat{\boldsymbol{x}}_*^{(h)}$, defined similar to the Eq. (3.14). The conditional distribution $p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right)$ follows the variational sparse GP predictive Eq. (2.27) and is itself given by:

$$\begin{aligned} p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right) &= \mathcal{N}\left(f_*^{(h)}\middle|\rho_*^{(h)}, \varsigma_*^{(h)}\right), \\ \rho_*^{(h)} &= \boldsymbol{k}_{*z}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)}, \\ \varsigma_*^{(h)} &= K_*^{(h)} - \boldsymbol{k}_{*z}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{k}_{z*}^{(h)} + \boldsymbol{k}_{*z}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{S}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{k}_{z*}^{(h)}, \end{aligned} \tag{3.29}$$

where the moments of $q\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)}\middle|\boldsymbol{m}^{(h)}, \boldsymbol{S}^{(h)}\right)$ are the optimal expressions derived in the Appendix A.2 (Eq, (A.18)) and repeated here for completeness:

$$\begin{aligned} \boldsymbol{S}^{(h)} &= \boldsymbol{K}_z^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\boldsymbol{K}_z^{(h)}, \\ \boldsymbol{m}^{(h)} &= \frac{1}{\sigma_h^2}\boldsymbol{K}_z^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^{\top}\boldsymbol{\mu}^{(h)}. \end{aligned}$$

The expectation in Eq. (3.28) is intractable, since the latent variables to be integrated appear in a complicated way inside the moments of the distribution in Eq (3.29). In order to proceed, we apply a Gaussian approximation, which implies computing only the first two moments of the predictive distribution. Those can be computed following properties of conditional distributions (GIRARD *et al.*, 2002):

$$\begin{aligned} p\left(f_*^{(h)}\right) &= \left\langle p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right)\right\rangle_{q(\boldsymbol{x}_*)} \approx \mathcal{N}\left(f_*^{(h)}\middle|\mu_*^{(h)}, \lambda_*^{(h)}\right), \\ \mu_*^{(h)} &= \left\langle \rho_*^{(h)}\right\rangle_{q(\boldsymbol{x}_*)}, \\ \lambda_*^{(h)} &= \left\langle \varsigma_*^{(h)}\right\rangle_{q(\boldsymbol{x}_*)} + \mathbb{V}_{q(\boldsymbol{x}_*)}\left\{\rho_*^{(h)}\right\}, \end{aligned}$$

where $\mathbb{V}_{q(\boldsymbol{x}_*)}\{\cdot\}$ indicates the variance operator with respect the distribution $q(\boldsymbol{x}_*)$. Importantly, the former expressions are analytical for the exponentiated quadratic covariance function.

Following the results derived by Quiñonero-Candela and Girard (2002), Quiñonero-

Candela *et al.* (2003), the predictive moments are finally given by:

$$p\left(f_*^{(h)}\right) = \left\langle p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right)\right\rangle_{q(\boldsymbol{x}_*)} \approx \mathcal{N}\left(f_*^{(h)}\middle|\mu_*^{(h)}, \lambda_*^{(h)}\right), \tag{3.30}$$

$$\mu_*^{(h)} = \left(\boldsymbol{B}^{(h)}\right)^\top \left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top, \tag{3.31}$$

$$\begin{aligned}\lambda_*^{(h)} = {} & \left(\boldsymbol{B}^{(h)}\right)^\top \left(\boldsymbol{\Psi}_{2*}^{(h)} - \left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top \boldsymbol{\Psi}_{1*}^{(h)}\right) \boldsymbol{B}^{(h)} + \Psi_{0*}^{(h)} \\ & - \mathrm{Tr}\left(\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1} - \left(\boldsymbol{K}_z^{(h)} + \sigma_h^{-2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\right)\boldsymbol{\Psi}_{2*}^{(h)}\right).\end{aligned} \tag{3.32}$$

Note that, since $x_*^{(h)} = f_*^{(h)} + \varepsilon_i^{(h)}$, following the RGP transition expression in Eq. (3.12), we have $q\left(x_*^{(h)}\right) = \mathcal{N}\left(x_*^{(h)}\middle|\mu_*^{(h)}, \lambda_*^{(h)} + \sigma_h^2\right)$. We have also defined the matrices

$$\boldsymbol{B}^{(h)} = \sigma_h^{-2}\left(\boldsymbol{K}_z^{(h)} + \sigma_h^{-2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top \boldsymbol{\mu}^{(h)}, \qquad 1 \leq h \leq H, \tag{3.33}$$

$$\boldsymbol{B}^{(H+1)} = \sigma_{H+1}^{-2}\left(\boldsymbol{K}_z^{(H+1)} + \sigma_{H+1}^{-2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top \boldsymbol{y}. \tag{3.34}$$

The terms $\Psi_{0*}^{(h)}$, $\boldsymbol{\Psi}_{1*}^{(h)}$ and $\boldsymbol{\Psi}_{2*}^{(h)}$ are computed as the original statistics in Eq. (3.27), but instead of the distributions $q\left(x_i^{(h)}\right)$ we use the new approximation $q\left(x_*^{(h)}\right)$ and replace $\boldsymbol{K}_f^{(h)}$ and $\boldsymbol{K}_{fz}^{(h)}$ respectively by $K_*^{(h)} = k\left(\hat{\boldsymbol{x}}_*^{(h)}, \hat{\boldsymbol{x}}_*^{(h)}\right)$ and $\boldsymbol{k}_{*z}^{(h)} = \left[k\left(\hat{\boldsymbol{x}}_*^{(h)}, \boldsymbol{\zeta}_1^{(h)}\right) \cdots k\left(\hat{\boldsymbol{x}}_*^{(h)}, \boldsymbol{\zeta}_M^{(h)}\right)\right]$. Note that for the observation layer we have $\mathbb{E}\{y_*\} = \mu_*^{(H+1)}$ and $\mathbb{V}\{y_*\} = \lambda_*^{(H+1)} + \sigma_{H+1}^2$.

It is worth emphasizing that the REVARB predictive expression in Eq. (3.30) is, as expected, recurrent, since future predictions are computed based on past predictive distributions. Such behavior is also made explicit in the original RGP join distribution in Eq. (3.19). Thus, it is clear that the RGP/REVARB framework is designed to simulate dynamical systems, with recurrences in both training and prediction steps always being operated with components of the own model, which are themselves probabilistically handled. This important feature turns our approach specially suitable to perform free simulation after the identification of nonlinear systems from noisy data.

Remark In rigorous terms, after each new prediction is made, the RGP model could be augmented with the newly computed latent variables $x_*^{(h)}$ and, since the model has changed, it could be re-optimized via the REVARB lower bound. Similar observation was also pointed out by Damianou *et al.* (2016) in the context of GPs with uncertain inputs. Of course, such procedure would be much more computationally demanding. We avoid this laborious approach and follow the more pragmatic alternative explained in this section, considering that the training data is enough to obtain a well tuned model and make all the predictions.

3.3.2 Sequential RNN-based Recognition Model

From the variational distribution $q(\boldsymbol{x}^{(h)})$ in Eq. (3.23) it is clear that the number of variational parameters in the REVARB framework grows linearly with the number of observed samples, which renders optimization challenging in large N scenarios. To alleviate this problem we propose an alternative to constrain the variational means $\boldsymbol{\mu}_i^{(h)}$ using recurrent neural networks (RNNs). More specifically, in the h-th hidden layer $(1 \leq h \leq H)$ we have:

$$\boldsymbol{\mu}_i^{(h)} = g_{\mu}^{(h)}\left(\hat{\boldsymbol{x}}_{i-1}^{(h)}\right) = \boldsymbol{\phi}_{\mu,L_N}\left(\boldsymbol{W}_{\mu,L_N}^{(h)\top}\boldsymbol{\phi}_{\mu,L_N-1}\left(\cdots\boldsymbol{\phi}_{\mu,1}\left(\boldsymbol{W}_{\mu,1}^{(h)}\hat{\boldsymbol{x}}_{i-1}^{(h)}\right)\right)\right), \tag{3.35}$$

where L_N denotes the depth of the neural network, matrices $\boldsymbol{W}_{\mu,l}^{(h)}\big|_{l=1}^{L_N}$ are the networks' weights and $\boldsymbol{\phi}_{\mu,l}(\cdot)\big|_{l=1}^{L_N}$ denote element-wise activation functions. Note that, since the neural network above is not a probabilistic model, the latent variables in its input $\hat{\boldsymbol{x}}_{i-1}^{(h)}$ are actually replaced by their correspondent variational mean values $\boldsymbol{\mu}_i^{(h)}$. The notation $\hat{\boldsymbol{x}}_{i-1}^{(h)}$ was maintained only as an analogy to the original RGP expression in Eq. (3.12).

We refer to this RNN-based constraint as the *sequential recognition model*. Such model directly captures the transition between the latent representation across time. This provides a constraint over the variational posterior distributions of the REVARB framework that maintains its emphasis in free simulation. The variational variances $\boldsymbol{\lambda}_i^{(h)}$ can be either modeled by a separate parametric model, with a distinct set of weights, or, for simplicity, be kept fixed to constant small values. For now we keep the latter simplification, since it was the one chosen to perform experiments in Section 3.4.5. Later, in Section 5.3.2 of Chapter 5, we deal with the case where both variational means and variances are modeled by NNs.

The recognition model's influence is combined with that of the analytic lower bound in the same objective optimization function. Thus, we no longer need to optimize the variational means but, instead, only the set of RNN weights, whose number does not increase linearly with N. Furthermore, this approach also allows us to kick-off optimization by random initialization of the RNN weights, as opposed to more elaborate initialization schemes for the variational parameters.

The recognition model idea relates to the work by Kingma and Welling (2014), Rezende *et al.* (2014). In our case, however, the recognition model is sequential to agree with the dynamical latent structure of the RGP model. Moreover, its purpose is distinct,

since it acts as a constraint in an already analytical variational lower bound. It is also important to emphasize that our sequential recognition model acts upon a nonparametric Bayesian model.

Although in this section we have started to introduce the sequential RNN-based recognition model, we will further discuss and detail it in Section 5.3.2 of Chapter 5, where it will be exploited in a stochastic optimization context in order to handle large scale datasets.

3.3.3 Multiple Inputs and Multiple Outputs

Almost all the examples and experiments in this thesis are related to single-input and single-output (SISO) systems. In the more general case of multiple-input and multiple-output (MIMO) data, the RGP/REVARB framework can still be applied, though with some minor modifications. We mention here two possible strategies, listed below.

Single shared latent space The simplest approach to apply the REVARB method to multiple-output data consists in considering the dynamics to be closely related among the outputs and to model them by a single shared dynamical latent space. In that case, we only need to modify the output layer of the RGP model by writing the observed outputs as a matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times D_y}$, a stack of N D_y-dimensional samples.

As follows we show the modified REVARB lower bound which is able to handle multiple outputs with this strategy, where, for the sake of simplicity, we have considered the same kernel hyperparameters and noise variance σ^2_{H+1} for all the output dimensions. We emphasize that we simply replicated the original bound in

Eq. (3.26) D_y times and replaced $\boldsymbol{y}$ by $\boldsymbol{y}_{:d}$, i.e., the d-th column of the matrix $\boldsymbol{Y}$:

$$
\begin{aligned}
\log p(\boldsymbol{Y}) &= \sum_{d=1}^{D_y} \log p(\boldsymbol{y}_{:d}), \\
\log p(\boldsymbol{Y}) &\geq \sum_{d=1}^{D_y} \Bigg\{ -\frac{N-L}{2} \sum_{h=1}^{H+1} \log 2\pi\sigma_h^2 \\
&\quad - \frac{1}{2\sigma_{H+1}^2} \left(\boldsymbol{y}_{:d}^\top \boldsymbol{y}_{:d} + \Psi_0^{(H+1)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)} \right)^{-1} \boldsymbol{\Psi}_2^{(H+1)} \right) \right) \\
&\quad + \frac{1}{2} \log \left| \boldsymbol{K}_z^{(H+1)} \right| - \frac{1}{2} \log \left| \boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2} \boldsymbol{\Psi}_2^{(H+1)} \right| \\
&\quad + \frac{1}{2(\sigma_{H+1}^2)^2} \boldsymbol{y}_{:d}^\top \boldsymbol{\Psi}_1^{(H+1)} \left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2} \boldsymbol{\Psi}_2^{(H+1)} \right)^{-1} \left(\boldsymbol{\Psi}_1^{(H+1)} \right)^\top \boldsymbol{y}_{:d} \\
&\quad + \sum_{h=1}^{H} \Bigg\{ -\frac{1}{2\sigma_h^2} \left(\sum_{i=L+1}^{N} \lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)} \right)^\top \boldsymbol{\mu}^{(h)} + \Psi_0^{(h)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)} \right)^{-1} \boldsymbol{\Psi}_2^{(h)} \right) \right) \\
&\quad + \frac{1}{2} \log \left| \boldsymbol{K}_z^{(h)} \right| - \frac{1}{2} \log \left| \boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2} \boldsymbol{\Psi}_2^{(h)} \right| \\
&\quad + \frac{1}{2(\sigma_h^2)^2} \left(\boldsymbol{\mu}^{(h)} \right)^\top \boldsymbol{\Psi}_1^{(h)} \left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2} \boldsymbol{\Psi}_2^{(h)} \right)^{-1} \left(\boldsymbol{\Psi}_1^{(h)} \right)^\top \boldsymbol{\mu}^{(h)} \\
&\quad - \sum_{i=1}^{N} \int_{x_i^{(h)}} q\left(x_i^{(h)} \right) \log q\left(x_i^{(h)} \right) + \sum_{i=1}^{L} \int_{x_i^{(h)}} q\left(x_i^{(h)} \right) \log p\left(x_i^{(h)} \right) \Bigg\} \Bigg\}.
\end{aligned}
\tag{3.36}
$$

Note that there is no increase in the number of variational parameters. Moreover, the increase in the computational cost is only moderate, since most of the terms in Eq. (3.36) are the same for all the output dimensions, with the exception of the ones in the output layer directly dependent of the vector $\boldsymbol{y}_{:d}$. This is the approach we have adopted in Sections 3.4.4 and 3.4.5 to model human motion data with 57 output dimensions.

Autoregressive MIMO An alternative to the single shared latent space consists in adapting the strategy usually applied to NARX and NARMAX MIMO models, described by Billings (2013). Besides altering the output layer, replacing the vector $\boldsymbol{y}$ by the columns of the matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times D_y}$ and using the modified bound presented in Eq. (3.36), we also need to adjust the definition of the regressors used as inputs in each layer. If we have D_u-dimensional inputs, stacked in the matrix $\boldsymbol{U} \in \mathbb{R}^{N \times D_u}$, and D_x-dimensional latent variables in each hidden layer h, stacked in the matrix

$\boldsymbol{X}^{(h)} \in \mathbb{R}^{N \times D_x}$, the regressors are computed by

$$\bar{\boldsymbol{u}}_i = [U_{i1}, \cdots, U_{1L_u}, \cdots, U_{d1}, \cdots, U_{dL_u}, \cdots, U_{D_u 1}, \cdots, U_{D_u L_u}]^\top, \tag{3.37}$$

$$\bar{\boldsymbol{x}}_i^{(h)} = [X_{i1}^{(h)}, \cdots, X_{1L}^{(h)}, \cdots, X_{d1}^{(h)}, \cdots, X_{dL}^{(h)}, \cdots, X_{D_x 1}^{(h)}, \cdots, X_{D_x}^{(h)}]^\top. \tag{3.38}$$

The variational distribution of the latent dynamical variables is now given by $q\left(\boldsymbol{x}_d^{(h)}\right) = \mathcal{N}\left(\boldsymbol{x}_d^{(h)} \middle| \boldsymbol{\mu}_d^{(h)}, \mathrm{diag}\left(\boldsymbol{\lambda}_d^{(h)}\right)\right)$, where $\boldsymbol{\mu}_d^{(h)} \in \mathbb{R}^N$, $\boldsymbol{\lambda}_d^{(h)} \in \mathbb{R}^N_{>0}$, $\boldsymbol{x}_d^{(h)}$ is the d-th column of $\boldsymbol{X}^{(h)}$ and $\mathbf{diag}(\cdot)$ builds a diagonal matrix from its argument.

The expressions in Eqs. (3.37) and (3.38) can be directly used to obtain the inputs defined in Eq. (3.14). Then, after the appropriate adjustments (replace $\boldsymbol{\mu}^{(h)}$ and $\lambda_i^{(h)}$ respectively by $\boldsymbol{\mu}_d^{(h)}$ and $[\boldsymbol{\lambda}_d^{(h)}]_i$), the modified MIMO bound in Eq. (3.36) can be computed. We note that this approach can become problematic if the dimensions D_u and D_x and lags of the regressors are too large.

3.3.4 Implementation Details

Algorithm 2 summarizes the application of the REVARB framework to perform inference with the RGP model in the context of dynamical modeling. The use of the algorithm itself is straightforward after implementing the bound in Eq. (3.26) and its analytical gradients with respect to the kernel and variational parameters.

Similar to most machine learning methods, it is convenient to follow some recommendations when implementing and using REVARB in practice. We list some of the implementation details we found more useful as follows.

Numerical stability Rasmussen and Williams (2006) mention many mathematical recommendations to maintain the numerical stability during GP model selection. We use most of their suggestions in the REVARB optimization step, especially the inclusion of a *jitter* term in the diagonal of the sparse covariance matrices $\boldsymbol{K}_z^{(h)}$ (more details below) and the use of Cholesky decomposition to perform both matrix inversions and computation of the log-determinants in the bound, since it is faster and more stable.

Another useful mathematical trick consists in applying the Woodbury matrix identity when the inversion of a possibly numerical unstable matrix is needed. The identity is given by:

$$(\boldsymbol{A} + \boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{C}^{-1} + \boldsymbol{VA}^{-1}\boldsymbol{U})^{-1}\boldsymbol{VA}^{-1}, \tag{3.39}$$

Algorithm 2: REVARB for dynamical modeling with the RGP model.

- Estimation step

Require: $\boldsymbol{u} \in \mathbb{R}^N$ (external input), $\boldsymbol{y} \in \mathbb{R}^N$ (output), H (number of hidden layers), M (number of inducing points), L (latent order lag), L_u (input order lag)

Initialize kernel hyperparameters and variational parameters;

repeat

Compute the evidence lower bound with Eq. (3.26);

Compute the analytical gradients of Eq. (3.26) with respect to the unknown parameters;

Update parameters with a gradient-based method (e.g. BFGS);

until convergence or maximum number or iterations

Output the optimized parameters;

- Free simulation with test data

Require: Test external inputs $\boldsymbol{u}_* \in \mathbb{R}^{N_*}$ and the previously estimated RGP model

for $i = 1 : N_*$ **do**

for $h = 1 : H$ **do**

Compute the predictive mean $\mu_{*i}^{(h)}$ and variance $\lambda_{*i}^{(h)}$ with Eqs. (3.31) and (3.32);

Update the variational distribution of the new latent dynamical variable with $q\left(x_{*i}^{(h)}\right) = \mathcal{N}\left(x_{*i}^{(h)} \middle| \mu_{*i}^{(h)}, \lambda_{*i}^{(h)} + \sigma_h^2\right)$;

end for

Compute the predictive mean $\mu_{*i}^{(H+1)}$ and variance $\lambda_{*i}^{(H+1)}$ of the output layer with Eqs. (3.31) and (3.32);

Output $y_{*i} \sim \mathcal{N}\left(\mu_{*i}^{(H+1)}, \lambda_{*i}^{(H+1)} + \sigma_{H+1}^2\right)$;

end for

where all the matrices have appropriate dimensions. For instance, if a problematic matrix $\boldsymbol{K}$ is written as $\boldsymbol{C} = \boldsymbol{K}^{-1}$, after applying the Woodbury formula we get the right hand side of Eq. (3.39), an expression where no inversions of the matrix $\boldsymbol{K}$ are necessary.

Finally, the REVARB lower bound in Eq. (3.26) contains the terms $\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)$, which appear inside an inversion and a log-determinant. Those operations can become unstable, so we apply the following transformation in each layer:

$$\begin{aligned}\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)} &= \boldsymbol{L}_z^{(h)}\left(\boldsymbol{L}_z^{(h)}\right)^\top + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)} \\ &= \boldsymbol{L}_z^{(h)} \underbrace{\left(\boldsymbol{I} + \frac{1}{\sigma_h^2}\left(\boldsymbol{L}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\left(\left(\boldsymbol{L}_z^{(h)}\right)^\top\right)^{-1}\right)}_{\boldsymbol{A}^{(h)}}\left(\boldsymbol{L}_z^{(h)}\right)^\top\end{aligned}$$

where $\boldsymbol{L}_z^{(h)} \in \mathbb{R}^{M\times M}$ is the Cholesky factor of $\boldsymbol{K}_z^{(h)}$. Since $\boldsymbol{L}_z^{(h)}$ is triangular, we can compute the terms that contain its inverse with fast and stable back-forward substitution. Furthermore, the auxiliary matrix $\boldsymbol{A}^{(h)}$ denoted above is usually much

more stable to invert or compute the log-determinant, using Cholesky decomposition again.

The jitter term In his technical report, Titsias (2009b) argues that the jitter term usually added to the diagonal of the sparse covariance matrix, e.g., $\boldsymbol{K}_z^{(h)} + \nu^{(h)}\boldsymbol{I}$ in the h-th layer of the RGP, is itself a variational parameter. He shows that its addition is equivalent to generalize the sparse variational bound by considering noisy inducing samples. In the case of the RGP model, the new inducing variables $\boldsymbol{z}'^{(h)} \in \mathbb{R}^M$ are given by:

$$\boldsymbol{z}'^{(h)} = \boldsymbol{z}^{(h)} + \boldsymbol{\varepsilon}_z^{(h)}, \quad \text{where } \boldsymbol{\varepsilon}_z^{(h)} \in \mathbb{R}^M \text{ and } \boldsymbol{\varepsilon}_z^{(h)} \sim \mathcal{N}(\boldsymbol{0}, \nu^{(h)}\boldsymbol{I}).$$

This assumption does not change the original lower bound if we already consider the jitters in the matrices $\boldsymbol{K}_z^{(h)}$. Since the jitter terms are variational parameters, they can be optimized along all the other parameters. Titsias mentions that a sensible strategy consists in initializing the jitter term with large values (we usually choose 10^{-2} or 10^{-3}), to improve numerical stability in the beginning of the optimization. After few iterations the jitter is automatically optimized to much lower values, usually around 10^{-6} or even lower, which is related to a tighter bound.

Model initialization REVARB is a deterministic approximation method, since there is no sampling or stochastic gradients in its optimization step. However, distinct initializations of the kernel and variational parameters can result in different local optima and affect the performance with test data[4]. We aim to alleviate this issue by sensibly initializing the model. For instance, the kernel hyperparameters are initialized as follows:

$$\begin{aligned}
[\sigma_f^2]^{(h)} &= \mathbb{V}\{\boldsymbol{\mu}^{(h)}\}, && 1 \leq h \leq H, \\
[\sigma_f^2]^{(H+1)} &= \mathbb{V}\{\boldsymbol{y}\}, \\
[w_d^2]^{(h)} &= C/(\max(\hat{\boldsymbol{x}}_{:d}^{(h)}) - \min(\hat{\boldsymbol{x}}_{:d}^{(h)}))^2, && 1 \leq h \leq H+1,\ 1 \leq d \leq D, \\
\sigma_h^2 &= 0.01\mathbb{V}\{\boldsymbol{\mu}^{(h)}\}, && 1 \leq h \leq H, \\
\sigma_{H+1}^2 &= 0.01\mathbb{V}\{\boldsymbol{y}\},
\end{aligned}$$

where $\hat{\boldsymbol{x}}_{:d}^{(h)} \in \mathbb{R}^N$ is the collection of the d-th component of all the h-th layer's inputs and C is a positive constant, usually in the range $[1,6]$ (a typical value is $C = 2$). We

[4]Note that different local optima can actually be related to distinct, but plausible, interpretations of the data. See Chapter 5 of Rasmussen and Williams (2006) for a discussion about local optima in the context of GP model selection.

also normalize the observations $\boldsymbol{y}$ and inputs $\boldsymbol{u}$ with zero mean and unitary standard deviation.

The initialization of the variational means $\boldsymbol{\mu}^{(h)}$ is made with information from outputs and inputs (when available). Thus, we initialize the means in all the hidden layers as follows:

$$\boldsymbol{\mu}^{(h)} = [\boldsymbol{y}, \boldsymbol{u}]\boldsymbol{p}_1,$$

where the vector $\boldsymbol{p}_1 \in \mathbb{R}^2$ is the principal component of the matrix $[\boldsymbol{y}, \boldsymbol{u}] \in \mathbb{R}^{N\times 2}$. Such initialization corresponds to the Principal Component Analysis (PCA) projection of the data in the dimension of $\boldsymbol{\mu}^{(h)}$. If the external inputs $\boldsymbol{u}$ are not available (e.g. for a time series), we simply initialize $\boldsymbol{\mu}^{(h)} = \boldsymbol{y}$. The variational variances are initialized with values in the interval $[0.01, 0.5]$. We usually choose $\boldsymbol{\lambda}_i^{(h)} = 0.2$, $\forall i$, in the recurrent layers.

Finally, the initialization of the pseudo inputs $\boldsymbol{\zeta}_j^{(h)}\big|_{j=1}^{M}$ is performed by applying a clustering algorithm to the inputs $\hat{\boldsymbol{x}}_i^{(h)}\big|_{i=1}^{N-L}$ of the layer h. We use the PAM (Partition Around Medoids) (KAUFMAN; ROUSSEEUW, 1990)[5] algorithm, more specifically, the deterministic R implementation available at the *cluster* package (MAECHLER *et al.*, 2016).

We note that some of those techniques were applied before to other variational GP-based models (HENSMAN *et al.*, 2013; DAMIANOU, 2015; BAUER *et al.*, 2016).

Optimization strategy It has been noted before that the optimization of variational bounds based on Titsias' sparse framework should be performed with care. For instance, in Damianou (2015) it is implied that the first optimization iterations should enforce a certain degree of SNR (Signal to Noise Ratio) by fixing the values of the kernel hyperparameters $[\sigma_f^2]^{(h)}$ and σ_h^2 to a reasonable initialization, such as the one mentioned in the previous item[6]. We also hold the value of the jitters $\boldsymbol{\nu}^{(h)}$, to reflect the fact that the initial iterations do not correspond to a tight bound.

In our experiments, we usually hold $[\sigma_f^2]^{(h)}$, σ_h^2 and $\boldsymbol{\nu}^{(h)}$ fixed in all layers for the first 100 iterations of the BFGS algorithm. Then, we unfix them and let the optimization

[5] PAM is an algorithm for the k-medoids clustering method, related to the more common K-means algorithm.

[6] The recommendation was emphasized in personal communication with Damianou.

procedure continue. Such strategy is important to avoid trivial bad local optima, e.g., to consider all data to be noise and make $[\sigma_f^2]^{(h)}$ too low or σ_h^2 too high.

Model orders As argued by Frigola-Alcade and Rasmussen (2013), the choice of autoregressive orders when applying GP-based models with ARD hyperparameters in the covariance function is not critical to prediction performance. Given that a relatively high order is chosen, the optimization of the ARD hyperparameters $w_d^{(h)}$ is able to select the relevant input dimensions to the inference. The only possible issue to selecting an order larger than the optimal one is the computational time penalty. The standard RGP formulation accepts vectors of regressors $\bar{\boldsymbol{u}}_i = [u_{i-1}, \cdots, u_{i-L_u}]^\top$ for the external inputs. Alternatively, we could fix $L_u = 1$ and consider only the previous input value, similar to regular GP-SSMs. In that case, the latent dynamical variables in the transition layers would be responsible to incorporate the dynamics of both inputs and outputs. In our preliminary experiments, neither approach was clearly superior to the other.

Free simulation As previously mentioned, free simulation within the REVARB framework is done by recursive computation of Eqs. (3.31) and (3.32). However, it is not clear if the transition noise variances $\sigma_h^2|_{h=1}^H$ should be added to the predictive variance in the hidden layers. In other words, we could do the recurrence with the variables $f_*^{(h)}$ or with its noisy version $x_*^{(h)}$. For instance, some experiments by Frigola-Alcade *et al.* (2014) switch off the transition noise during the test step. However, we have noticed in our experiments that by doing so the prediction often becomes overconfident, i.e., the predicted output variance is too small. On the other hand, by including the propagation of the transition noises we are able to better estimate the variance in the output. We follow the latter strategy, which almost always does not harm the predictive means and can even slightly improve them.

3.4 Experiments

In this section we evaluate the performance of our RGP model with REVARB inference in the tasks of nonlinear system identification, times series simulation and human motion modeling.

Quantitative evaluation is done by calculating the root mean squared error (RMSE) of the free simulation on the test data (not used in the training step),

given by $\mathbf{RMSE} = \sqrt{\frac{1}{N_*}\sum_{i=1}^{N_*}(y_i - \tilde{\mu}_i)^2}$, where N_* is the number of test samples, y_i is the true test output and $\tilde{\mu}_i$ is the predicted mean output. In some experiments we also compute the average negative log-predictive density (NLPD), given by $\mathbf{NLPD} = \frac{1}{2}\log 2\pi + \frac{1}{2N_*}\sum_{i=1}^{N_*}\left[\log \tilde{\sigma}_i^2 + \frac{(y_i-\tilde{\mu}_i)^2}{\tilde{\sigma}_i^2}\right]$, where $\tilde{\sigma}_i^2$ is the i-th predicted variance. The NLPD is a metric of the type "the lower, the better" and is useful to indicate a favorable balance between the squared error and the compactness of the uncertain prediction.

3.4.1 Initial Example

Before presenting the main experiments of this chapter, it is useful to show some simpler application of the RGP/REVARB framework in order to have a better initial intuition of its functioning. We use an artificial dynamical system modified from an example presented by Kocijan (2016), named simply as *Example* dataset and described by the following set of equations:

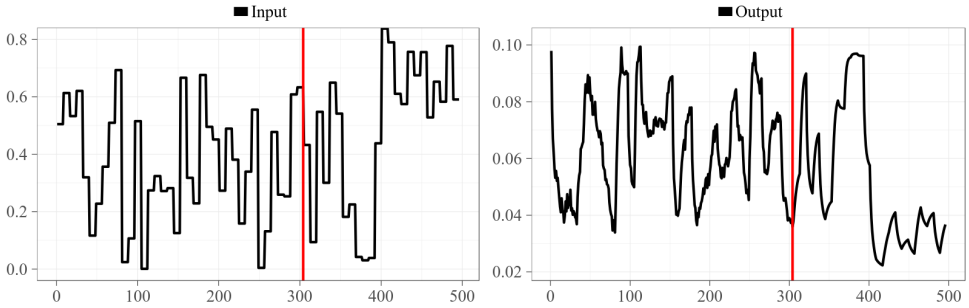
$$x_i^{(1)} = x_{i-1}^{(1)} + 0.5\frac{x_{i-1}^{(1)}x_{i-1}^{(2)}}{x_{i-1}^{(1)} + x_{i-1}^{(2)}} - 0.5u_{i-1}x_{i-1}^{(1)}, \tag{3.40}$$

$$x_i^{(2)} = x_{i-1}^{(2)} - 0.5\frac{x_{i-1}^{(1)}x_{i-1}^{(2)}}{x_{i-1}^{(1)} + x_{i-1}^{(2)}} - 0.5u_{i-1}x_{i-1}^{(2)} + 0.05u_{i-1}, \tag{3.41}$$

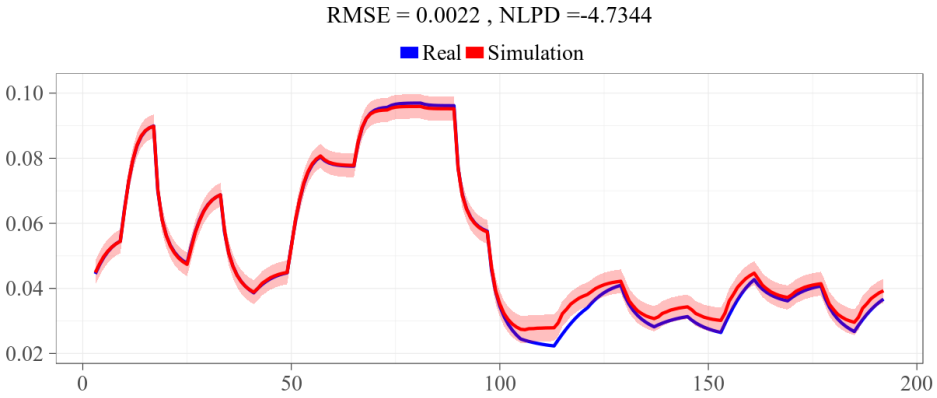
$$y_i = x_i^{(1)} + \mathcal{N}(0, 0.00165^2) \tag{3.42}$$

A total of 496 samples were generated, 304 for training and 192 for testing. The training data used 8-iteration long steps as inputs, each one scaled by uniformly random samples in the interval $[0, 0.7]$. The testing data was obtained from similar steps, but we allowed some larger scalings at the second half of the test set (up to 0.85). The generated dataset is illustrated in Figs. 14 a) and b), where it is possible to notice the presence of some test inputs at the end of the sequence which are in a range not covered by the training data.

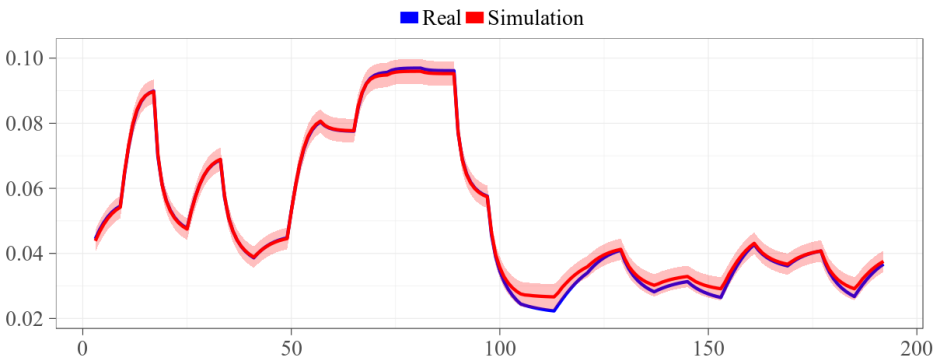
We emphasize that the free simulation evaluation is made based only on the test inputs and past predictions. We trained a standard GP-NARX and two RGP models, one with $H = 1$ and other with $H = 2$ hidden layers, using orders $L = L_u = 2$ and $M = 30$ pseudo-inputs. The obtained results are presented in Figs. 14 c) to e), where the shaded areas indicate ± 2 standard deviations around the predicted means. All the models were able to learn the dynamics from the training set, since the first half test simulation, which closely follows the training data, indicate predictions very close to the correct output.



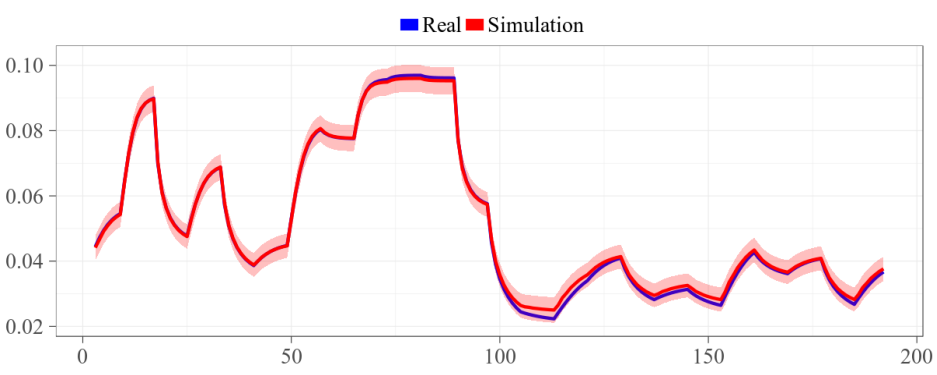
a) Input data. b) Output data.



c) GP-NARX.



d) RGP ($H = 1$).



e) RGP ($H = 2$).

Figure 14 – Example of system identification with the RGP model. The presented free simulations were generated from the test data. The vertical **red** line in a) and b) separates the training (left) from the test data (right).

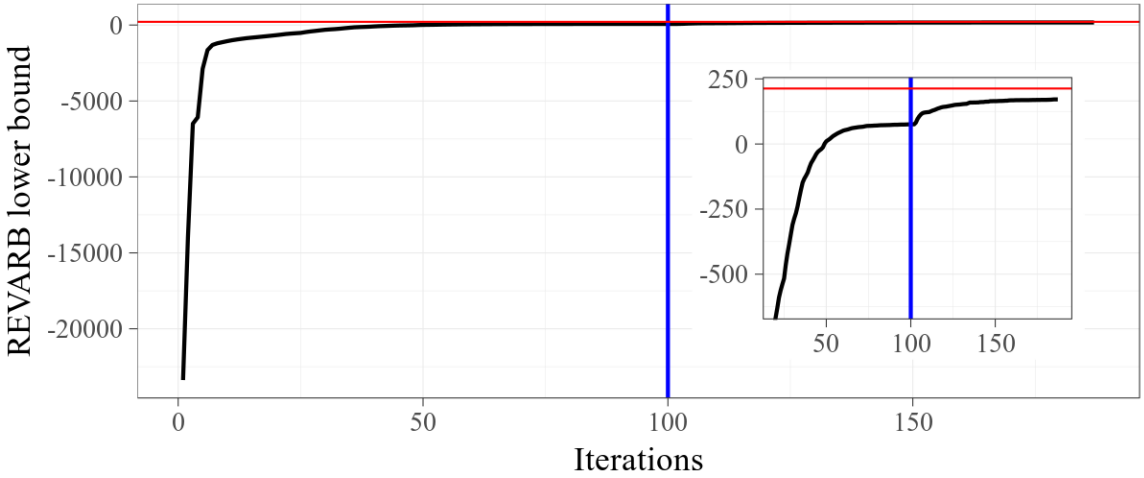


Figure 15 – Convergence curve of the REVARB lower bound during the training step of the RGP model with $H = 2$ hidden layers on the *Example* dataset using the BFGS algorithm. The vertical **blue** line indicates the point where the noise variance hyperparameters are unfixed (see Section 3.3.4) and the smaller picture is a zoomed version of the curve after the first 20 iterations. The horizontal **red** line indicates the analytical marginal log-likelihood obtained by the standard GP-NARX model.

However, the second half of the test data, which contains inputs outside the training range, was much better handled by the RGP models, especially the one with two hidden layers, which not only presented simulations closer to the desired signal but also better indicated the uncertainty around its predictions, resulting in improved RMSE and NLPD values.

The optimization of the RGP model with $H = 2$ hidden layers, which contains a total of 1543 variational parameters and kernel hyperparameters, took 187 iterations of the algorithm BFGS, following the initialization and optimization procedures explained in Section 3.3.4. The convergence curve of the REVARB lower bound is illustrated in Fig. 15, where the vertical blue line indicates the instant where the noise variance hyperparameters are unfixed and the horizontal red line indicates the analytical marginal log-likelihood obtained by the GP-NARX model.

Although it is not so easy to interpret the numerical difference between the REVARB lower bound and the GP-NARX marginal log-likelihood, variational theory proves that such gap is proportional to the Kullback-Leibler divergence between the variational posterior Q and the true posterior, as presented in Section 2.6.1. Moreover, the obtained non-zero divergence value is due to the approximations assumed by the REVARB framework in order to get analytical expressions. Nevertheless, from the results illustrated in Fig. 14 we can see that such gap, and hence the considered approximations, had no

affect on the estimation procedure of the RGP model, which indicates that the REVARB lower bound acts as a good enough proxy for the true marginal log-likelihood.

3.4.2 Nonlinear System Identification

We now reproduce and update the results with the system identification task firstly reported in our work Mattos *et al.* (2016), comprised by one artificial benchmark, introduced by Narendra and Li (1996), and two real datasets. The *Artificial* dataset is given by the following expressions:

$$x_i^{(1)} = \frac{x_{i-1}^{(1)}}{(1+(x_{i-1}^{(1)})^2)+1}\sin(x_{i-1}^{(2)}), \tag{3.43}$$

$$x_i^{(2)} = x_{i-1}^{(2)}\cos(x_{i-1}^{(2)}) + x_{i-1}^{(1)}\exp\left[-\frac{1}{8}((x_{i-1}^{(1)})^2+(x_{i-1}^{(2)})^2)\right] + \frac{u_{i-1}^3}{1+u_{i-1}^2+0.5\cos(x_{i-1}^{(1)}+x_{i-1}^{(2)})}, \tag{3.44}$$

$$y_i = \frac{x_i^{(1)}}{1+0.5\sin(x_i^{(2)})} + \frac{x_i^{(2)}}{1+0.5\sin(x_i^{(1)})}, \tag{3.45}$$

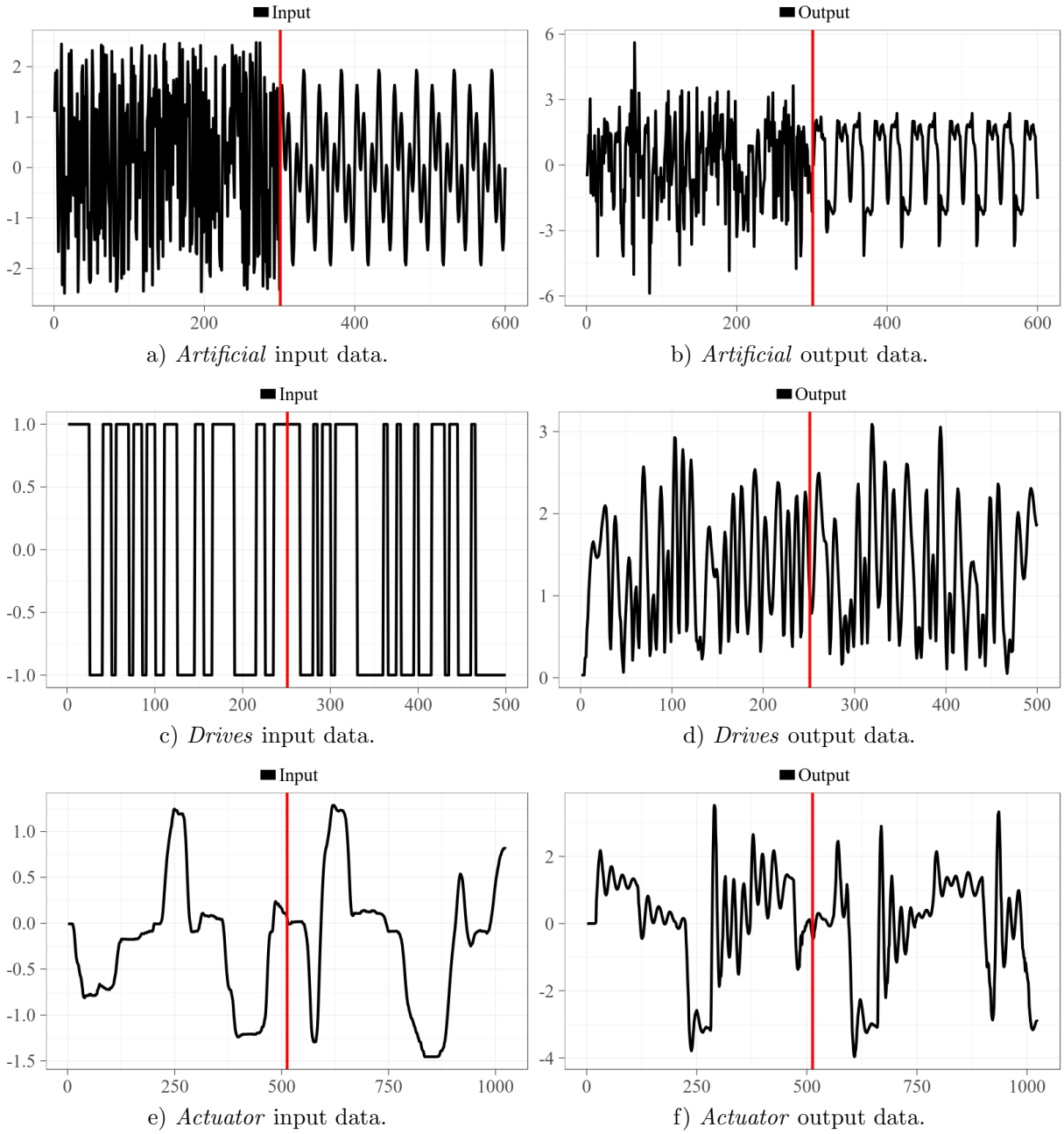
where u_i and y_i are respectively the external input and output in the i-th instant.

The first real dataset, labeled *Drives* and introduced by Wigren (2010)[7], consists of 500 samples from a system with two electric motors that drive a pulley using a flexible belt. The input is the sum of voltages applied to the motors and the output is the speed of the belt. The second dataset, labeled *Actuator* and described by Sjöberg *et al.* (1995)[8], consists of 1024 samples from a hydraulic actuator that controls a robot arm, where the input is the size of the actuator's valve opening and the output is its oil pressure.

In the case of the *Artificial* dataset we choose $L = L_u = 5$ and generate 300 samples for training and 300 samples for testing, using the same inputs described by Narendra and Li (1996), i.e., $u_i = \mathrm{U}(-2.5, 2.5)$ for training, which indicates samples from a uniform distribution in the interval $[-2.5, 2.5]$, and $u_i = \sin(2\pi i/10) + \sin(2\pi i/25)$ for testing. We also added to the outputs of the training data noise sampled from $\mathcal{N}(0, 0.1)$. For the real datasets we use $L = L_u = 10$ and apply the first half of the data for training and the second one for testing. Fig. 16 illustrates the input and output series related to the considered datasets.

[7] Data available at <http://www.it.uu.se/research/publications/reports/2010-020/NonlinearData.zip>. We used input u1 and output z1.

[8] Data available at <http://www.iau.dtu.dk/nnbook/systems.html>.



a) *Artificial* input data. b) *Artificial* output data.

c) *Drives* input data. d) *Drives* output data.

e) *Actuator* input data. f) *Actuator* output data.

Figure 16 – Datasets considered for the nonlinear system identification task. The vertical **red** lines separate the training (left) from the test data (right). We note that in the case of the *Artificial* dataset the training and test data were not actually generated one after the other.

We apply our RGP model with 2 hidden layers, REVARB inference and 30 pseudo-inputs for the *Artificial* dataset and 50 pseudo-inputs for *Drives* and *Actuator* datasets. We compare it with two models commonly applied to system identification tasks: standard GP-NARX and MLP-NARX. We use the MLP implementation from the MATLAB Neural Network Toolbox with 1 hidden layer (MATLAB, 2013). We also include experiments with the LSTM network, although the task itself probably does not require

Table 2 – Summary of RMSE values for the free simulation results on system identification test data.

	Artificial	*Drives*	*Actuator*
MLP-NARX	1.6334	0.4403	0.4621
LSTM	2.2438	0.4329	0.5170
GP-NARX	1.9245	0.4128	1.5488
RGP ($H = 2$)	**0.4513**	**0.1922**	**0.3104**

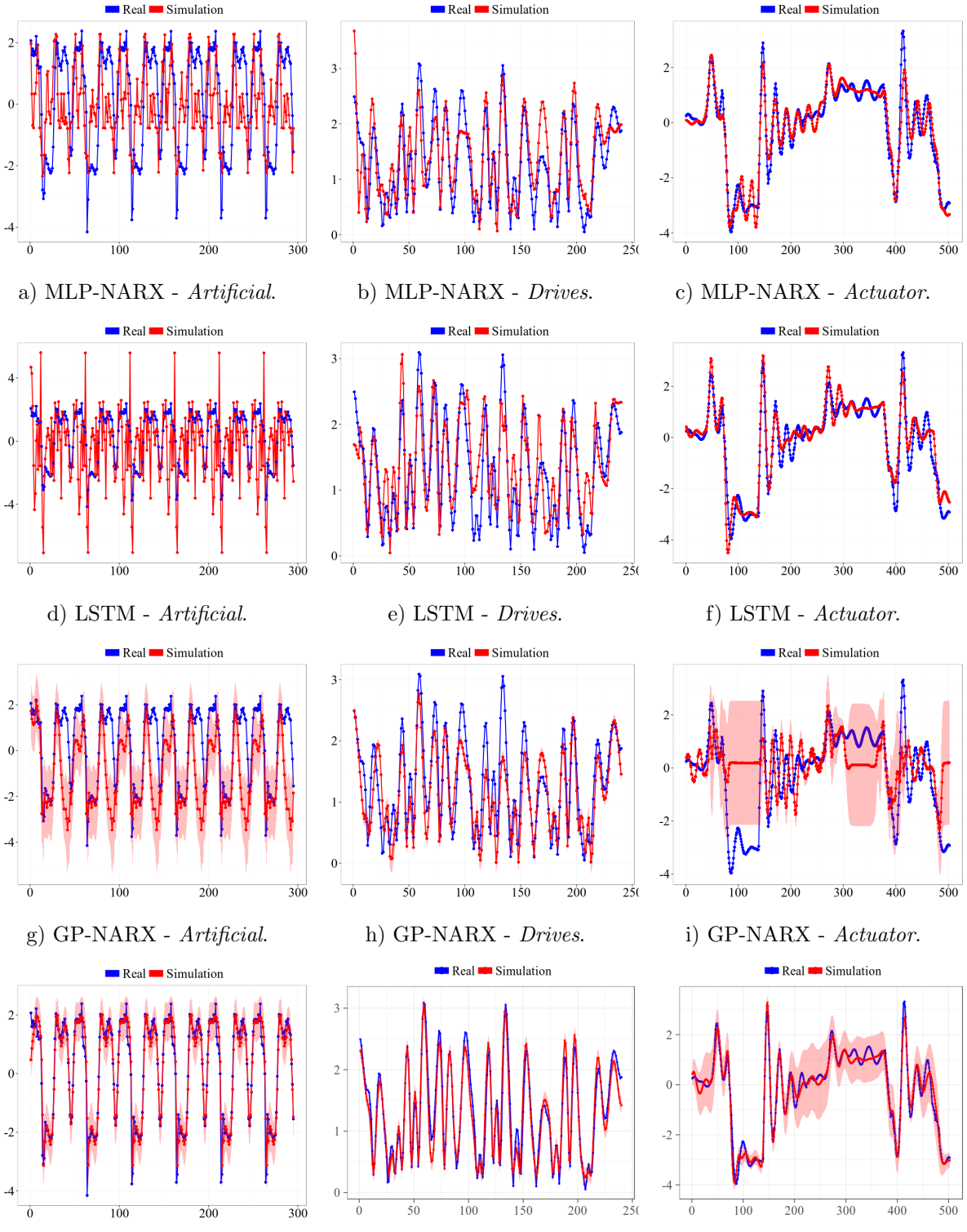
long term dependences. The original LSTM architecture by Hochreiter and Schmidhuber (1997) was chosen, with a network depth of 1 to 3 layers and the number of cells at each layer selected to be up to 2048. LSTM memory length was unlimited, and sequence length was chosen initially to be a multiple of the longest duration memory present in the data generative process and reduced gradually.

The obtained RMSE values are summarized in Tab. 2 and the obtained simulations are illustrated in Fig. 17, with shaded area indicating ± 2 standard deviations around the predicted means in the GP-based cases. The RGP model was superior in all experiments, with large improvements over GP-NARX. It is interesting to also notice, especially for the *Artificial* and *Actuator* datasets, how the RGP model was able to better indicate the larger uncertainty in the regions with more difficult predictions. Although worse than RGP, the MLP-NARX model presented a relatively good result for the *Actuator* dataset. The higher RMSE values obtained by the LSTM model is possibly related to the difficulties encountered when trying to optimize its architecture for this given task.

We note that some RGP results in Tab. 2 are updated from the values originally reported in Mattos *et al.* (2016), since we have incorporated some of the implementation details discussed in Section 3.3.4. In the recent work by Al-Shedivat *et al.* (2016), a new deep recurrent kernel structure named GP-LSTM is proposed and evaluated with the datasets *Drives* and *Actuator*. The authors obtain respectively the RMSE values 0.225 and 0.347, which are better then the results originally reported in Mattos *et al.* (2016) but slightly worse than our most updated results presented in Tab. 2.

3.4.2.1 Magneto-Rheological Fluid Damper Data

We now evaluate the RGP model in the task of modeling the nonlinear dynamics of a Magneto-Rheological (MR) Fluid Damper, an experiment described by Wang *et al.*



a) MLP-NARX - *Artificial*. b) MLP-NARX - *Drives*. c) MLP-NARX - *Actuator*.

d) LSTM - *Artificial*. e) LSTM - *Drives*. f) LSTM - *Actuator*.

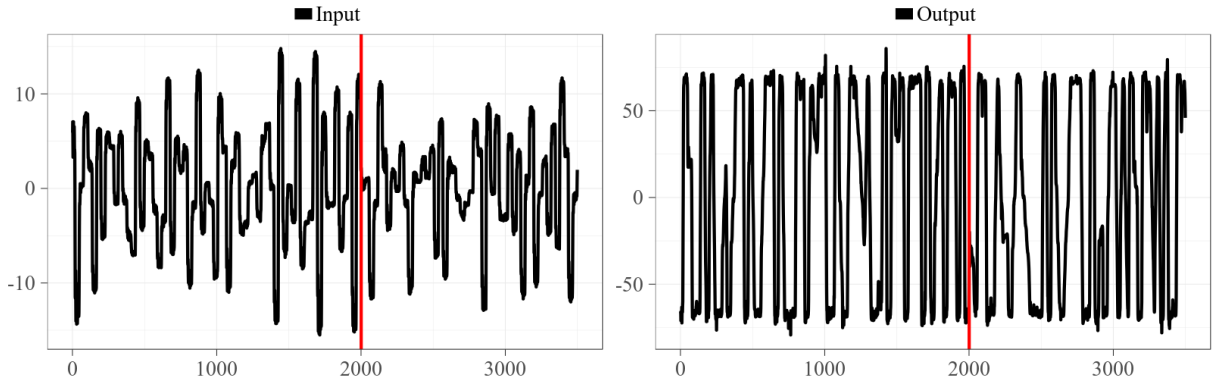
g) GP-NARX - *Artificial*. h) GP-NARX - *Drives*. i) GP-NARX - *Actuator*.

j) RGP - *Artificial*. k) RGP - *Drives*. l) RGP - *Actuator*.

Figure 17 – Free simulation on nonlinear system identification test data. In the GP-based models the shaded areas indicate ± 2 standard deviations around the predicted mean values.

(2009)[9]. The damper is a semi-active control device to reduce vibrations on dynamical

[9]Data available in the System Identification Toolbox for Mathworks Matlab (The MathWorks Inc., 2016).



a) Input data. b) Output data.

Figure 18 – Input and output series for the *Damper* dataset. The vertical **red** lines separate the training (left) from the test data (right).

Table 3 – RMSE and NLPD values for the free simulation results on the *Damper* dataset. The first five experiments were reported by The MathWorks Inc. (2016). The reduced-rank GP-SSM experiment was presented by Svensson *et al.* (2016). The experiment with SISOG was reported by Bijl *et al.* (2016). The NA entry indicates that the NLPD value was not reported.

	RMSE	NLPD
Linear OE model (4th order)	27.1	-
Hammerstein-Wiener (4th order)	27.0	-
NARX (3rd order, wavelet)	24.5	-
NARX (3rd order, Tree partition)	19.3	-
NARX (3rd order, sigmoid network)	8.24	-
Standard GP-NARX	13.31	13.71
Variational Sparse GP-NARX ($M = 100$)	13.83	14.44
Reduced-rank GP-SSM (SVENSSON *et al.*, 2016)	8.17	3.71
SISOG (BIJL *et al.*, 2016)	7.12	NA
RGP ($H = 1$)	11.18	3.47
RGP ($H = 2$)	**6.04**	**3.05**

structures. The variable viscosity of the MR fluid is used to control the damping force. The dataset consists of inputs related to the system velocity and outputs related to the achieved damping force for a given fluid viscosity.

The same set-up presented by Svensson *et al.* (2016) is applied, which consists of using the first 2000 of the original 3499 data samples for training and the remaining for test via free simulation. Fig. 18 illustrates the input and output of the *Damper* dataset.

We used in the experiments two RGP models, with 1 and 2 hidden layers, the orders $L = L_u = 3$ and $M = 100$ pseudo-inputs. The obtained RMSE and NLPD results are presented in the last row of Tab. 3, along with the results for some more conventional system identification methods reported by The MathWorks Inc. (2016). We also include the recent result obtained by a reduced-rank GP-SSM, which applies a particle MCMC

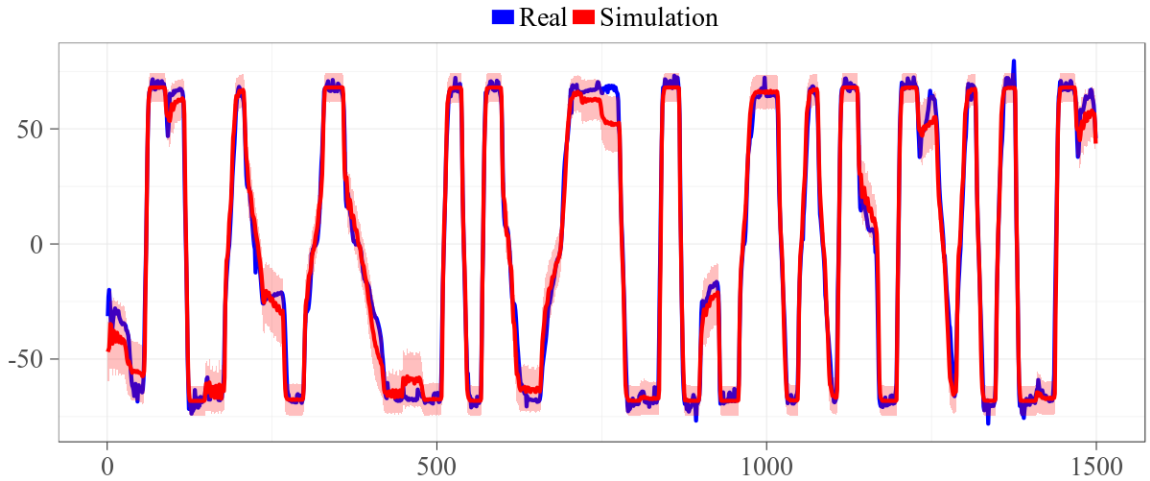


Figure 19 – Free simulation on test data with the 2-hidden layer RGP model after estimation on the *Damper* dataset. The shaded areas indicate ±2 standard deviations around the predicted means.

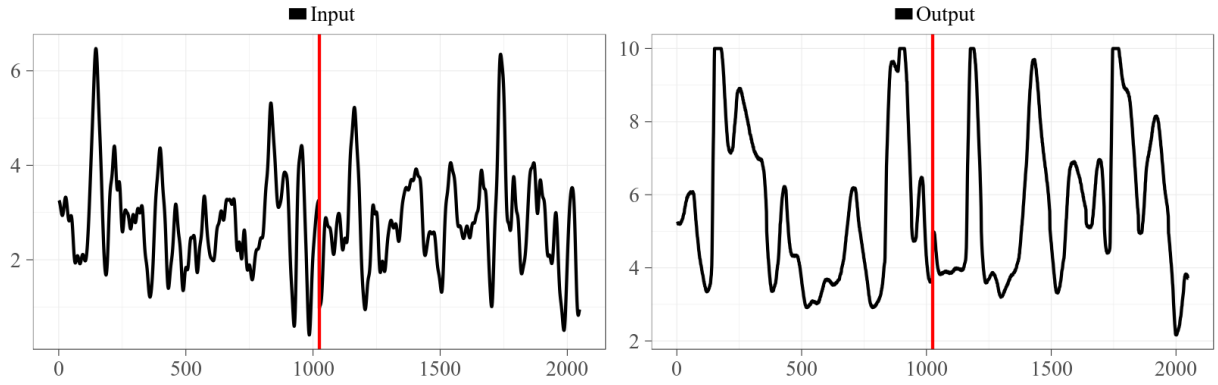
strategy to perform Bayesian inference, reported by Svensson *et al.* (2016), and the result obtained by Bijl *et al.* (2016) with the SISOG (System Identification using Sparse Online GP) algorithm.

Our RGP/REVARB approach with 2 transition layers outperformed all the other models, also presenting better compactness, as indicated by its lower NLPD value. Fig. 19 illustrates the free simulation on test data with the 2-hidden layer RGP model, where the shaded areas indicate ±2 standard deviations around the predicted means. Once again our model was able to provide larger predicted variances, indicating greater uncertainty, only in the regions where its predictions differ more from the real data.

3.4.2.2 Cascaded Tanks Data

Schoukens *et al.* (2015)[10] provide a dataset collected from a physical system consisting of two cascaded water tanks to be used in dynamical modeling benchmarks. The input signal controls a water pump that pumps the water to a reservoir into the upper water tank. Then, the water flows through a small opening to the lower tank and afterwards back to the reservoir through another small opening. The output is the measured flow from the lower tank into the reservoir. It is mentioned that one of the challenges in this set-up is the eventual occurrence of overflows in the upper tank due to large input values, which is related to hard saturation nonlinearities and input dependent noise. The dataset, comprised of 1024 training samples and 1024 test samples, is illustrated

[10]Data available at <http://homepages.vub.ac.be/~mschouke/benchmark2016.html>.



a) Input data. b) Output data.

Figure 20 – Input and output series for the *Cascaded Tanks* dataset. The vertical **red** lines separate the training (left) from the test data (right).

in Fig. 20.

Svensson and Schön (2017) consider such *Cascaded Tanks* dataset to evaluate several system identification methods, including one proposed by the authors, a Bayesian model with basis function expansion with a connection to GP-SSMs where inference is performed by SMC. Those results reported by Svensson and Schön (2017) are presented in Tab. 4. We also include the recent result reported by Schoukens and Scheiwe (2016) with a nonlinear Volterra feedback model and the metrics obtained by our RGP model with both 1 and 2 transition layers, orders $L_u = 5$ ($L_u = 1$ for the model with 2 hidden layers) and $L = 5$ and $M = 50$ pseudo-inputs. The 2-hidden layer RGP model, whose free simulation on test data is presented in Fig. 21, obtained the best RMSE value. Although this time the RGP simulation seems overconfident, since the predicted variances are clearly too small (note that the 2-hidden layers version did not present the best NLPD value), we emphasize that we were not able to find in the recent literature a simulation RMSE better than the one achieved by our model for this particular dataset.

3.4.3 Time Series Simulation

Although the focus of the present thesis is on system identification tasks, which are usually related to some dynamical data generated by a sequence of known external inputs, the RGP model and the REVARB inference framework can also be directly applied to the problem of forecasting time series by simply omitting the exogenous input u_i and the related variables.

As an illustration, we consider the standard benchmark of the Mackey-Glass chaotic time series, which is defined by the differential equation below (MACKEY *et al.*,

Table 4 – Free simulation results on the *Cascaded Tanks* dataset. The first six experiments were reported by Svensson and Schön (2017). The Volterra model result was reported by Schoukens and Scheiwe (2016). The NA entry indicates that the NLPD value was not reported.

	RMSE	NLPD
Linear SSM (2nd order)	0.67	-
NARX (sigmoid network, 5th order)	0.73	-
NARX (sigmoidnet, 5th order, simulation focus)	0.49	-
NARX (wavelet network, 5th order)	0.61	-
NARX (wavelet network, 5th order, simulation focus)	0.64	-
GP-SSM with basis function expansion (SVENSSON; SCHÖN, 2017)	0.45	NA
Standard GP-NARX	1.50	1079
Variational Sparse GP-NARX ($M = 50$)	0.5040	119.3
Nonlinear Volterra feedback model (SCHOUKENS; SCHEIWE, 2016)	0.3972	-
RGP ($H = 1$)	0.7973	**2.3295**
RGP ($H = 2$)	**0.3084**	7.793

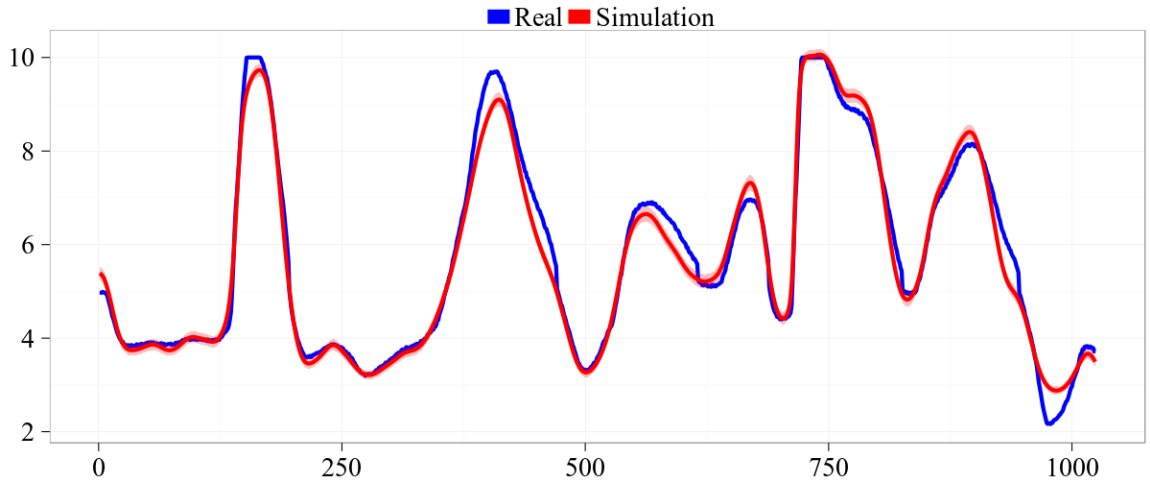


Figure 21 – Free simulation on test data with the RGP model after estimation on the *Cascaded Tanks* dataset. Although the simulation is good, the shaded areas indicating ±2 standard deviations around the predictions are too small (almost not visible in the figure), which suggests that the model is overconfident.

1977):

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = -By_t + A\frac{y_{t-T}}{1+y_{t-T}^C}, \tag{3.46}$$

where t represents the discrete time. We use the following standard values for the constants in the equation: $A = 0.2$, $B = 0.1$, $C = 10$ and $T = 17$.

We consider the same set-up used by Damianou (2015), where 72 noiseless samples were generated for training and free simulation is performed for the next 1110 iterations. Data was normalized with zero mean and unitary variance. We use a RGP model with 1 hidden layer, order $L = 18$ and $M = 30$ pseudo-inputs. The RMSE and

Table 5 – Results for free simulation on the Mackey-Glass time series. With the exception of the RGP model, all experiments were performed and the results reported by Damianou (2015). The NA entries indicate the unreported NLPD values.

	RMSE	NLPD
Standard GP-NARX	1.08	NA
GP-NARX with uncertainty propagation (GIRARD *et al.*, 2002)	0.956	NA
Semi-described GP-NARX (DAMIANOU; LAWRENCE, 2015)	0.742	NA
RGP ($H = 1$)	**0.547**	25.89

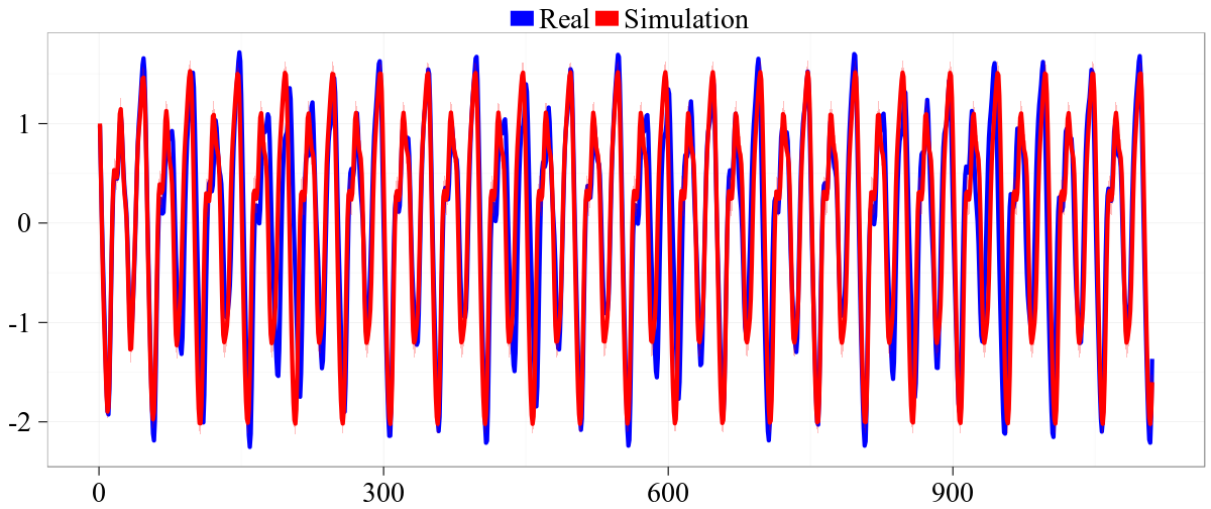


Figure 22 – Free simulation of the Mackey-Glass chaotic time series with the RGP model. Note that the predicted uncertainty is too small and barely visible at some points, indicating overconfident predictions.

NLPD values for our model is presented in Tab. 5. We also include the RMSE results reported by Damianou (2015) with an autoregressive GP model with uncertain inputs (semi-described GP-NARX, introduced by Damianou and Lawrence (2015)), the standard GP-NARX and the GP-NARX with propagation of uncertainty via moment matching (proposed by Girard *et al.* (2002) and experimented by Damianou).

The RGP model, whose free simulation output is shown in Fig. 22, performed considerably better than the other methods. Although the predictions are clearly overconfident (the predicted uncertainty is barely visible in some points), we can verify that even after several hundreds of iterations, based solely on past predictions, the simulation holds close to the real time series.

Table 6 – Summary of RMSE values for the free simulation results on human motion test data with an included external control input given by the y coordinate of the left toes. Training and test sets present both walking and running motions.

MLP-NARX	GP-NARX	RGP ($H = 2$)
1.2141	0.8987	**0.8600**

3.4.4 Human Motion Modeling

We now reproduce the experiments presented in Mattos *et al.* (2016) with the motion capture data from the CMU database[11] to model walking and running motions with the RGP/REVARB framework. Training was performed with the trajectories 1 to 4 (walking) and 17 to 20 (running) from subject 35. The test set is comprised by the trajectories 5 to 8 (walking) and 21 to 24 (running) from the same subject. The original dataset contains 59 outputs, but 2 are constant, so we remove those and use the remaining 57. We follow the single shared latent space strategy for multiple output modeling, summarized in Section 3.3.3.

In order to perform free simulation with an external signal, we include a control input given by the y coordinate of the left toes. Following the previous system identification experiments, predictions are made based only on such control input and previous predictions. We also normalize the inputs and outputs with zero mean and unitary standard deviation before training.

We evaluate a 2-hidden layers RGP with 200 pseudo-inputs, the standard GP-NARX model and a 1-hidden layer MLP with 1000 hidden units. The orders are fixed at $L = L_u = 20$. Note that the data related to both walking and running is used in the same training step. RGP's latent autoregressive structure allow us to train a single model for all outputs (see Section 3.3.3). In the case of GP-NARX, we had to train separate models for each output, since training a single model with 57 (output dimensions) $\times 20$ $(L) + 20$ $(L_u) = 1160$ dimensional regressor vector was not feasible.

Test RMSE values are summarized in Tab. 6. The RGP model obtained better results than both the other models. We emphasize that RGP has an additional advantage over GP-NARX due to its latent autoregressive structure, which allows the training of a single model for all the 57 outputs.

[11] Data available at <http://mocap.cs.cmu.edu/>.

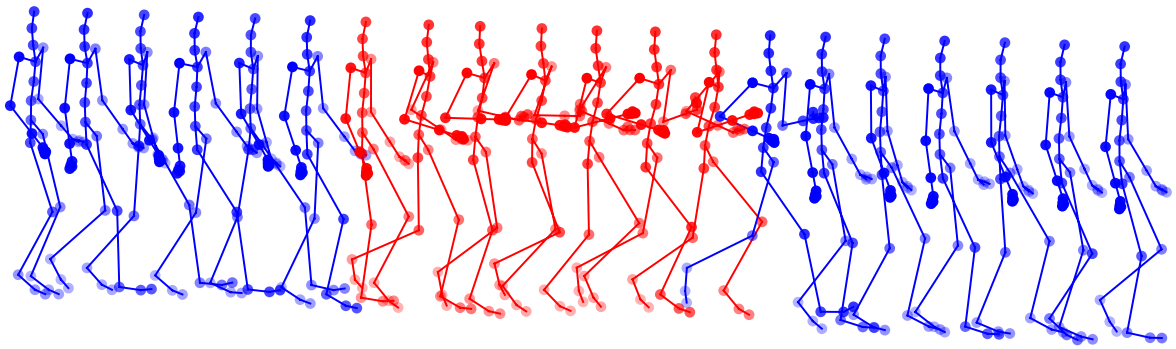


Figure 23 – Motion generated by the RGP model with a step function control signal for the average velocity, starting with walking (**blue**), switching to running (**red**) and switching back to walking (**blue**)

3.4.5 Avatar Control

We conclude this experiments section by applying the RGP model to the synthesis of human motions with simple control signals, such as the velocity, also reported in Mattos *et al.* (2016). This methodology can ideally be used to generate realistic human motion according to human instruction in virtual environment, such as video games. We use the 5 walking and 5 running sequences from CMU motion database described in previous experiment and take the subject's average velocity as the control signal. We then train a 1-hidden layer RGP model with the RNN sequential recognition model (two hidden layers with 500-200 units). After training, we use the model to synthesize motions with unseen control signals.

Fig. 23 shows the frames of the generated motion with a step function signal. We emphasize that the training sequences do not contain any switch of motions, which forced the model to interpolated the different velocity regimes. Videos of some of the generated motions are available at <https://youtu.be/FuF-uZ83VMw>, <https://youtu.be/FR-oeGxV6yY> and <https://youtu.be/AT0HMtoPgjc>.

3.5 Discussion

In this section we have defined the broad family of Recurrent Gaussian Processes (RGPs) models, which, similarly to other recurrent modeling strategies, such as RNNs, are able to learn, possibly deep, temporal representations from data. Our novel RGP model presents internal dynamics in the form of a latent autoregressive structure. The intractabilities brought by the recurrent GP priors are tackled via a variational

approximation approach, resulting in the REVARB framework. Furthermore, we extended REVARB with a sequential RNN-based recognition model that simplifies the optimization in some learning scenarios (which will be more explored later in Chapter 5).

We applied the RGP/REVARB framework to the tasks of nonlinear system identification, times series forecasting and human motion modeling. The good results obtained by our model indicate that the latent autoregressive structure and our variational approach were able to better capture the dynamical behavior of the data when compared to the other evaluated learning methods.

In this chapter we only applied models with up to 2 hidden layers, which actually results in a model with 3 GP priors (2 transition layers and 1 observation layer). From previous experiences with non-recurrent deep GPs, it has been noted that due to the strong nonlinearities and nonparametric expressiveness of hierarchical GP models, one typically needs fewer layers than, for instance, when using deep neural networks, even for complex data (DAMIANOU, 2015). Thus, it is usually observed that the addition of a layer to a shallow GP has a much greater effect than adding a layer to a NN, given that, as presented in Section 2.5, one GP layer is equivalent to a 1-hidden layer NN with infinite hidden units. In that sense, the proposed RGP approach is a new step towards a similar relation between GPs and RNNs. Still, investigations with datasets that require deeper models is left for future work.

In the work by Turner and Sahani (2008), the authors present some concerns with respect to the use of mean-field approximations within a time-series context, suggesting that such approximation has a hard time propagating uncertainty through time. However, we observed in practice that our proposed REVARB framework is able to better account for uncertainty in the latent space with its autoregressive deep structure. This may be due to the next layer being able to "compensate" the mean-field assumption of the previous layer, accounting for additional (temporal) correlations. Since each latent variable $x_i^{(h)}$ and, thus, its associated variational parameters, is present in exactly two layers (see Eq. (3.14), which details each layer's input), such effect is enabled for all latent variables of the model. Moreover, since the pseudo-inputs in a given layer are shared among the different dynamical latent variables in that layer, they induce correlations between them. Related remarks were made for regular deep GPs by Damianou (2015).

Nevertheless, it is worth noting that in some experiments, such as the one with

the *Cascaded Tanks* dataset, in Section 3.4.2.2, and the one with the Mackey-Glass times series, in Section 3.4.3, the RGP model was not able to output acceptable error bars, i.e., the predicted variances were clearly too small. Although we cannot state categorically the cause for that, we suspect that it is an optimization issue. For instance, if some pseudo-inputs stay far from the training data even after the optimization step, they could falsely indicate lower uncertainty in that area during predictions. Further tuning of the initialization and optimization methodologies presented in Section 3.3.4 may overcome such behavior.

In the next chapter we will continue to explore the use of GP-based models for dynamical learning, but in scenarios where the assumption of Gaussian observation noise is not acceptable, more specifically when it is expected the presence of outliers in the estimation data.

4 ROBUST BAYESIAN MODELING WITH GP MODELS

"If you do not expect the unexpected,
you will not recognize it when it arrives."
(Heraclitus of Ephesus)

As mentioned in the previous chapters, GP models are nonparametric data-driven techniques, where instead of a rigid prespecified structure, the model allows for the data to "speak by itself". Moreover, a typical GP model increases its complexity as more data becomes available and estimation data is used to both optimize the model and make predictions.

Despite these appealing features, standard GP models assume a Gaussian observation noise and, hence, Gaussian likelihoods arise naturally within the framework. While this is not an issue for many applications, it makes the difference in modeling scenarios contaminated with non-Gaussian noise, such as impulsive noise, commonly treated as a type of *outlier*. Such estimation samples containing outliers can result in misestimated hyperparameters during training and directly affect predictions. In that case, the model's performance considerably deteriorates, compromising its generalization capability.

The definition of an outlier varies in the literature. One could intuitively state that it consists of a data point which significantly differs from the overall observed data (AGGARWAL, 2013). In other words, one could label as outlier those observations which deviates so much from the remaining data that it suggests to have been generated by a different mechanism (HAWKINS, 1980). Thus, it is important to emphasize that one observation can only be termed as an outlier with respect to a given generative assumption. In this work, such interpretation is relative to the common Gaussianity assumption considered for the noise model.

This chapter aims to address the issue of training GP-based models that are able to learn from data contaminated with non-Gaussian noise in the form of outliers. Following the general trend in the present thesis, our focus lies in the modeling of data related to dynamical systems.

We pursue models that are able to directly learn dynamics from data containing outliers, as opposed to methodologies which aim to remove them from the data before

training, i.e., a *data cleaning* step (PEARSON, 2002). Such outlier detection approach, surveyed in the context of temporal data by Gupta *et al.* (2014), can be useful when a single or few outliers are present, but the diagnosis becomes much more difficult when dealing with multiple outliers (ROUSSEEUW; LEROY, 2005) which are intermingled with the samples of the system under analysis.

The *robustness* that we seek is related to models that are not strongly affected by the presence of outliers in the training data, a valuable feature for system identification methods (MILANESE *et al.*, 2013). Huber, a pioneer in robust statistics, also argues against the data cleaning approach, emphasizing that directly dealing with outliers avoids erroneous removal of potentially valuable training samples (HUBER, 2011).

In such context, we will evaluate some of the state-of-the-art GP models for robust regression in the task of system identification and also propose two robust GP-based dynamical models specifically designed to tackle that class of problems. We conclude the chapter by evaluating the described models with several experiments related to robust nonlinear system identification.

4.1 Robust GP Model with Non-Gaussian Likelihood

It is well known that the standard GP model with Gaussian likelihood is not robust to outliers due to its light tails. The standard Bayesian approach to tackle impulsive noise considers a heavy-tailed distribution for the likelihood of the N observed data samples $\boldsymbol{y} \in \mathbb{R}^N$ given the latent function values $\boldsymbol{f} \in \mathbb{R}^N$, such as a mixture of Gaussians, the Laplace or the Student-t, which are respectively given by (GELMAN *et al.*, 2014a):

$$p_{\text{Mix}}(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{N} \left[(1-w_o)\mathcal{N}(y_i|f_i,\sigma^2) + w_o\mathcal{N}(y_i|f_i,\sigma_o^2) \right], \tag{4.1}$$

$$p_{\text{Lap}}(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{N} \frac{1}{2s} \exp\left(-\frac{|y_i - f_i|}{s} \right), \tag{4.2}$$

$$p_{\text{Stu}}(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{N} \frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)\sqrt{\pi\nu\sigma^2}} \left(1 + \frac{1}{\nu}\frac{(y_i - f_i)^2}{\sigma^2} \right)^{-(\nu+1)/2}, \tag{4.3}$$

where w_o, σ^2, σ_o^2, s and ν are likelihood hyperparameters and $\Gamma(\cdot)$ is the gamma function. The first distribution has two components, one Gaussian with σ^2 variance to model the regular samples and another Gaussian with σ_o^2 variance to model the discrepant values. Note that $\sigma_o^2 \gg \sigma^2$ and that w_o regulates the mixture ratio. On the other hand, the Laplace

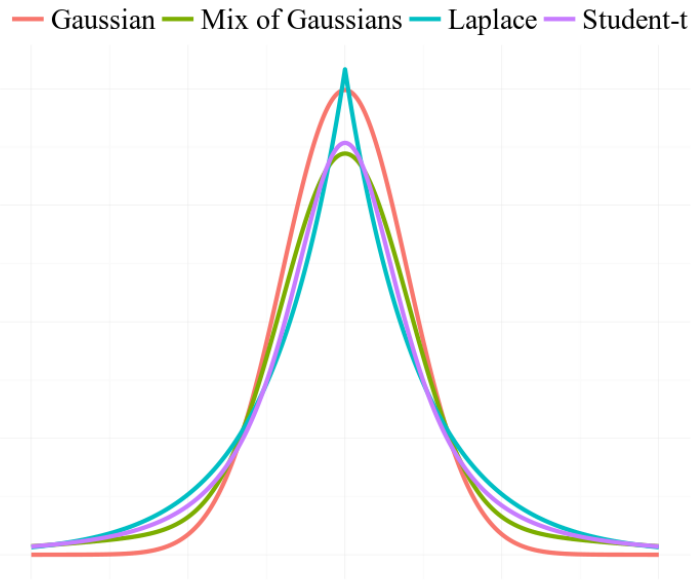
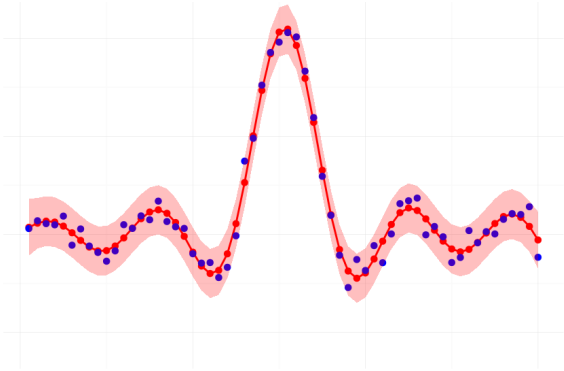


Figure 24 – Comparison between the Gaussian likelihood and heavy-tailed distributions. Note that the presented non-Gaussian distributions allow some probability mass for values far from the mean.

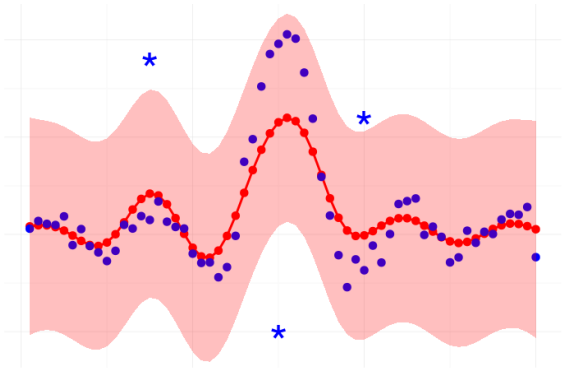
and Student-t expressions explicitly define distributions that prevent tails' probability from decaying too fast.

Fig. 24 illustrates a comparison between the curves related to those heavy tailed distributions and the Gaussian likelihood, where we can see that the formers allow some probability mass for samples far from the mean. Such behavior results in robustness, since it prevents the model from "overadapting" itself in order to support the outliers. This effect can be seen in Fig. 25, which shows a simple regression problem with the normalized *sinc* function ($f_i = \frac{\sin(\pi x_i)}{\pi x_i}$) in the presence of outliers tackled by the standard GP with Gaussian likelihood and a robust GP with Student-t likelihood. Note that the predicted mean in the robust version is barely affected by the outliers in the data.

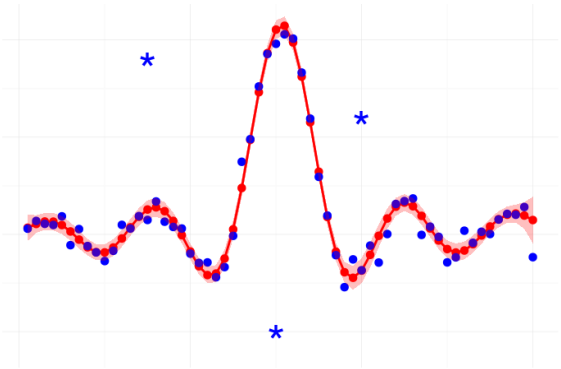
However, once a non-Gaussian likelihood is chosen, many of the original GP equations become intractable (non-analytical), as can be recalled from the Gaussian properties used throughout the Chapter 2 to obtain analytical expressions. Thus, robust GP models require the use of approximation methods, such as Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) (NEAL, 1997; BOTTEGAL *et al.*, 2014; KANTAS *et al.*, 2015; SCHÖN *et al.*, 2015), variational Bayes (VB) (JORDAN *et al.*, 1999; WAINWRIGHT *et al.*, 2008; BLEI *et al.*, 2017) and expectation propagation (EP) (MINKA, 2001; GELMAN *et al.*, 2014b), where MCMC and SMC follow stochastic sampling techniques and VB and EP are deterministic approximations. In this thesis we



(a) Standard GP with Gaussian noise.



(b) Standard GP deteriorated by outliers.



c) Robust GP with Student-t likelihood.

Figure 25 – Effect of outliers on both standard and robust GP regression models.

will cover the use of VB and EP for robust GP learning, which will be described shortly.

Outlier-robust GP models for regression have been well covered in the literature by works that apply Student-t likelihood (NEAL, 1997; JYLÄNKI *et al.*, 2011; BERGER; RAUSCHER, 2012), Laplace likelihood (KUSS, 2006) and mixture of Gaussians (KUSS *et al.*, 2005; NAISH-GUZMAN; HOLDEN, 2008; STEGLE *et al.*, 2008) to handle outliers as non-Gaussian noise. Robust classification with GPs has also been studied before (KIM; GHAHRAMANI, 2008; HERNÁNDEZ-LOBATO *et al.*, 2011). However, developing such GP models specifically for robust dynamical system identification is a relatively new topic, with few representations besides our own work (MATTOS *et al.*, 2015; MATTOS *et al.*, 2016; MATTOS *et al.*, 2017). For instance, Bottegal *et al.* (2014) propose a robust GP model for linear impulse response identification with MCMC-based inference, but do not cover nonlinear systems. More recently, Ranjan *et al.* (2016) introduce an Expectation Maximization (EM) algorithm to tackle robust regression tasks, reporting an experiment with system identification, though it is evaluated in one-step-ahead prediction scenarios only.

As expected, dynamical modeling can also be compromised by the presence

of outliers in the estimation data. Actually, in autoregressive approaches, e.g., NARX models, the presence of outliers becomes even more problematic, since the noisy outputs, possibly containing outliers, are fed back to the inputs as regressors, greatly interfering in the model capability to learn the underlying dynamics of the system.

Thus, as follows we will describe two common approaches for robust regression with GPs, named by us the GP-tVB model, which applies VB to perform inference with a Student-t likelihood, and the GP-LEP model, which uses an EP algorithm and the Laplace likelihood. We emphasize that those models still follow the standard GP-NARX structure with regressors comprised of past outputs and exogenous inputs, as presented in Section 3.1.1. The difference lies in their noise assumptions, which now consider heavy-tailed likelihoods for the observations.

Though not with those names, both aforementioned robust GP models are extensively detailed within the robust regression context in the thesis by Kuss (2006), so our presentation will be brief. Our contribution consists in afterwards training those models following the GP-NARX formulation and evaluating them in the task of system identification in the presence of outliers.

Remark The robust Bayesian approach we follow, i.e., handling outliers as noise samples from a heavy tailed distribution, differs from frequentist approaches. For instance, robust M-estimation methods, as presented by Huber (2011), aim to use alternative cost functions in order to give smaller error penalties to the samples associated with large errors, supposedly related to outliers, which results in less model adaptation to corrupted observations.

4.1.1 The GP-tVB Model

Instead of directly applying the Student-t probability density function (Eq. (4.3)) in the likelihood, Kuss (2006) exploits the fact that it can be defined as a mixture of infinitely many Gaussian distributions with gamma distributed precisions (i.e., inverse variances)[1], sometimes called the *scale-mixture representation*:

$$\mathscr{T}(\varepsilon|\nu,\mu=0,\sigma)=\frac{\Gamma((\nu+1)/2)}{\Gamma(\nu/2)\sqrt{\pi\nu\sigma^2}}\left(1+\frac{\varepsilon^2}{\nu\sigma^2}\right)^{-(\nu+1)/2}, \tag{4.4}$$

$$\mathscr{T}(\varepsilon|\nu=2\alpha,\mu=0,\sigma=\sqrt{\beta/\alpha})=\int_0^\infty \mathscr{N}(\varepsilon|0,\tau^{-1})\Gamma(\tau|\alpha,\beta)\mathrm{d}\tau, \tag{4.5}$$

[1] In Kuss (2006) an inverse gamma prior was chosen for the variance of the Gaussian distribution, which is equivalent to the analysis presented here.

where zero mean Gaussians with τ precision were considered and α and β are respectively the *shape* and *inverse scale* hyperparameters of the gamma distribution. The resulting Student-t distribution for the variable ε, which also has mean $\mu = 0$, is characterized by the *degrees of freedom* ν and the *scale* hyperparameter σ. Lower values of ν result in heavier tails, while for $\nu \to \infty$ the tails become lighter and the distribution converges to a Gaussian.

Kuss applied the aforementioned strategy to provide a GP model with a Student-t likelihood as follows:

$$p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}_f), \tag{4.6}$$

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\tau}^{-1}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \mathrm{diag}(\boldsymbol{\tau}^{-1})), \tag{4.7}$$

$$p(\boldsymbol{\tau}|\alpha, \beta) = \prod_{i=1}^{N} \Gamma(\tau_i|\alpha, \beta), \tag{4.8}$$

where $\boldsymbol{K}_f \in \mathbb{R}^{N\times N}$ is the covariance matrix obtained from the matrix $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ of stacked inputs $\boldsymbol{x}_i|_{i=1}^{N}$, $\boldsymbol{y} \in \mathbb{R}^N$ is the vector of observations, $\boldsymbol{f} \in \mathbb{R}^N$ and $\boldsymbol{\tau} \in \mathbb{R}^N_{>0}$ are latent (unobserved) variables, $\mathbf{diag}(\cdot)$ builds a diagonal matrix from a vector and the precisions τ_i have a gamma prior with hyperparameters α and β. Note that the variances $\sigma_i^2 = \tau_i^{-1}$ are inverse gamma distributed. Such approach has also been used before for instance by Tipping and Lawrence (2005) to perform robust Bayesian interpolation with the Student-t likelihood.

In a VB context, the joint posterior of $\boldsymbol{f}$ and $\boldsymbol{\tau}$ is approximated by a factorized expression as follows:

$$p(\boldsymbol{f}, \boldsymbol{\tau}|\boldsymbol{y}, \boldsymbol{X}) \approx Q = q(\boldsymbol{f})q(\boldsymbol{\tau}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A}) \prod_{i=1}^{N} \Gamma(\tau_i|a_i, b_i), \tag{4.9}$$

where $q(\boldsymbol{f})$ and $q(\boldsymbol{\tau})$ are variational distributions and $\boldsymbol{m} \in \mathbb{R}^N$, $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^N_{>0}$ are unknown variational parameters.

A lower bound to the marginal log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{X})$ can be found relating it to the factorized posterior $q(\boldsymbol{t})q(\boldsymbol{\tau})$ by using Jensen's inequality (TIPPING; LAWRENCE, 2005; KUSS, 2006):

$$p(\boldsymbol{y}|\boldsymbol{X}) = \int_{\boldsymbol{f},\boldsymbol{\tau}} p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\tau}^{-1}) p(\boldsymbol{f}|\boldsymbol{X}) p(\boldsymbol{\tau}),$$

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \int_{\boldsymbol{f},\boldsymbol{\tau}} q(\boldsymbol{f})q(\boldsymbol{\tau}) \log \frac{p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\tau}^{-1}) p(\boldsymbol{f}|\boldsymbol{X}) p(\boldsymbol{\tau})}{q(\boldsymbol{f})q(\boldsymbol{\tau})}. \tag{4.10}$$

Kuss (2006) details how the optimal values of $\boldsymbol{m}$ and $\boldsymbol{A}$ in Eq. (4.9) can be written in terms of $\boldsymbol{a}$ and $\boldsymbol{b}$, which themselves are optimized, along with the kernel hyperparameters, with the help of the analytical gradients of the bound in Eq. (4.10). Such maximization is equivalent to minimize the Kullback-Leibler divergence $\mathrm{KL}(q(\boldsymbol{f})q(\boldsymbol{\tau})||p(\boldsymbol{f},\boldsymbol{\tau}|\boldsymbol{y},\boldsymbol{X}))$ between the variational distribution and the true posterior, which improves the approximation (TIPPING; LAWRENCE, 2005).

The optimization of the hyperparameters and the latent variables can be done in an Expectation-Maximization (EM) fashion, following the procedure described by Kuss (2006). Then, the moments of the Gaussian approximated prediction $p(f_*|\boldsymbol{y},\boldsymbol{X},\boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*,\sigma_*^2)$ for a new input $\boldsymbol{x}_*$ are given by

$$\mu_* = \boldsymbol{k}_{*f}(\boldsymbol{K}_f + \boldsymbol{\Sigma})^{-1}\boldsymbol{y}, \quad \text{and} \quad \sigma_*^2 = K_* - \boldsymbol{k}_{*f}(\boldsymbol{K}_f + \boldsymbol{\Sigma})\boldsymbol{k}_{f*}, \tag{4.11}$$

where $\boldsymbol{\Sigma} = \mathrm{diag}(\boldsymbol{b}/\boldsymbol{a})$, $\boldsymbol{k}_{*f} = [k(\boldsymbol{x}_*,\boldsymbol{x}_1),\cdots,k(\boldsymbol{x}_*,\boldsymbol{x}_N)]$, $\boldsymbol{k}_{f*} = \boldsymbol{k}_{*f}^\top$ and $K_* = k(\boldsymbol{x}_*,\boldsymbol{x}_*)$.

4.1.2 The GP-LEP Model

Kuss (2006) describes a GP model with a Laplace likelihood, as expressed in Eq. (4.2). Interestingly, such distribution can also be written as a mixture of Gaussians, this time with exponentially distributed variances:

$$\mathscr{L}(\varepsilon|\mu = 0, s) = \frac{1}{2s}\exp\left(-\frac{|\varepsilon|}{s}\right), \tag{4.12}$$

$$\mathscr{L}(\varepsilon|\mu = 0, s = 1/\sqrt{2\beta}) = \int_0^\infty \mathcal{N}(\varepsilon|0,\sigma^2)\mathrm{Exponential}(\sigma^2|\beta)\mathrm{d}\sigma^2, \tag{4.13}$$

where zero mean Gaussians with σ^2 variances were considered and β is the *rate* hyperparameter of the exponential distribution. The resulting Laplace distributed noise variable ε has mean $\mu = 0$ and is characterized by the *scale* hyperparameter s.

Kuss argues that a variational approximation for such model, similar to the one presented in Section 4.1.1 for the Student-t likelihood, would not be possible due to the exponential prior given to the variance σ^2 in Eq. (4.13), since the expectation $\mathbb{E}\{\sigma^{-2}\}$ for the precision does not have a finite value. Therefore, Kuss pursues an EP inference method to tackle the inference problem.

The EP algorithm usually works by approximating the true posterior distribution of $\boldsymbol{f} \in \mathbb{R}^N$, the vector of latent function values, by a Gaussian which follows a

factorized structure (MINKA, 2001; KUSS, 2006):

$$p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X}) = \frac{\mathcal{N}(\boldsymbol{f}|\boldsymbol{0},\boldsymbol{K}_f)}{p(\boldsymbol{y}|\boldsymbol{X})} p(\boldsymbol{y}|\boldsymbol{f},s) = \frac{\mathcal{N}(\boldsymbol{f}|\boldsymbol{0},\boldsymbol{K}_f)}{p(\boldsymbol{y}|\boldsymbol{X})} \prod_{i=1}^{N} p(y_i|f_i,s), \tag{4.14}$$

$$p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X}) \approx \frac{\mathcal{N}(\boldsymbol{f}|\boldsymbol{0},\boldsymbol{K}_f)}{q(\boldsymbol{y}|\boldsymbol{X})} \prod_{i=1}^{N} c(f_i,\mu_i,\sigma_i^2,Z_i) = q(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{m},\boldsymbol{A}), \tag{4.15}$$

where the first expression comes from the straightforward application of the Bayes' rule and $c(f_i,\mu_i,\sigma_i^2,Z_i) = Z_i\mathcal{N}(f_i|\mu_i,\sigma_i^2)$ are called *site functions*, which include the normalizers Z_i. The scalar variables μ_i, σ_i^2 and Z_i are collectively called *site parameters*. Following Kuss (2006), the mean vector $\boldsymbol{m} \in \mathbb{R}^N$ and covariance matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ of the approximate distribution may be computed as $\boldsymbol{m} = \boldsymbol{A}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and $\boldsymbol{A} = (K^{-1} + \boldsymbol{\Sigma}^{-1})^{-1}$, where $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1^2,\cdots,\sigma_N^2)$ and $\boldsymbol{\mu} = [\mu_1,\cdots,\mu_N]^\top$. Moreover, the approximation $q(\boldsymbol{y}|\boldsymbol{X})$ (Eq. (4.15)) to the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{X})$ is shown to be expressed in terms of the site parameters and can be used to perform model selection.

Kuss (2006) details an EP algorithm to perform inference with a GP model equipped with a Laplace likelihood, which we name henceforth the GP-LEP model. The site parameters are optimized by iterative moment matching, which turns out to be equivalent to simultaneously minimize the reverse Kullback-Leibler divergence between the true posterior and the approximate distribution, i.e., the divergence $\mathrm{KL}(p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})||q(\boldsymbol{f}))$. The convergence is not guaranteed, but it has been reported in the literature that EP works well within GP models (RASMUSSEN; WILLIAMS, 2006).

Predictions $p(f_*|\boldsymbol{y},\boldsymbol{X},\boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*,\sigma_*^2)$ for a new input $\boldsymbol{x}_*$ are given by

$$\mu_* = \boldsymbol{k}_{*f}\boldsymbol{K}_f^{-1}\boldsymbol{m}, \quad \text{and} \quad \sigma_*^2 = K_* - \boldsymbol{k}_{*f}(\boldsymbol{K}_f^{-1} - \boldsymbol{K}_f^{-1}\boldsymbol{A}\boldsymbol{K}_f^{-1})\boldsymbol{k}_{f*}. \tag{4.16}$$

4.1.3 Evaluation of GP-tVB and GP-LEP for Robust System Identification

In order to verify the performance of the previously described models in the task of nonlinear system identification in the presence of outliers, we performed computational experiments with five artificial datasets, detailed in Tab. 7. The first four datasets were presented in the seminal work by Narendra and Parthasarathy (1990). The fifth dataset was generated following Kocijan *et al.* (2005).

Besides the Gaussian noise, indicated in the last column of Tab. 7, the estimation data of all datasets was also incrementally corrupted with a number of outliers equal to 2.5%, 5% and 10% of the estimation samples. Each randomly chosen sample was

Table 7 – Details of the five artificial datasets used in the computational experiments related to the task of robust system identification. The indicated noise in the last column is added only to the output of the estimation data. Note that $\mathrm{U}(A,B)$ is a random number uniformly distributed between A and B.

		Input/Samples		
#	**Output**	**Estimation**	**Test**	**Noise**
1	$y_i = \frac{y_{i-1}y_{i-2}(y_{i-1}+2.5)}{1+y_{i-1}^2+y_{i-2}^2} + u_{i-1}$	$u_i = \mathrm{U}(-2,2)$ 300 samples	$u_i = \sin(2\pi i/25)$ 100 samples	$\mathscr{N}(0,0.29)$
2	$y_i = \frac{y_{i-1}}{1+y_{i-1}^2} + u_{i-1}^3$	$u_i = \mathrm{U}(-2,2)$ 300 samples	$u_i = \sin(2\pi i/25) + \sin(2\pi i/10)$ 100 samples	$\mathscr{N}(0,0.65)$
3	$y_i = 0.8y_{i-1} + (u_{i-1}-0.8)u_{i-1}(u_{i-1}+0.5)$	$u_i = \mathrm{U}(-1,1)$ 300 samples	$u_i = \sin(2\pi i/25)$ 100 samples	$\mathscr{N}(0,0.07)$
4	$y_i = 0.3y_{i-1} + 0.6y_{i-2} + 0.3\sin(3\pi u_{i-1}) + 0.1\sin(5\pi u_{i-1})$	$u_i = U(-1,1)$ 500 samples	$u_i = \sin(2\pi i/250)$ 500 samples	$\mathscr{N}(0,0.18)$
5	$y_i = y_{i-1} - 0.5\tanh(y_{i-1} + u_{i-1}^3)$	$u_i = \mathscr{N}(u_i\|0,1)$ $-1 \leq u_i \leq 1$ 150 samples	$u_i = \mathscr{N}(u_i\|0,1)$ $-1 \leq u_i \leq 1$ 150 samples	$\mathscr{N}(0,0.0025)$

added by a uniformly distributed value $\mathrm{U}(-M_y,+M_y)$, where M_y is the maximum absolute output. We emphasize that only the output values were corrupted in this step. Such outlier contamination methodology is similar to the one performed by Majhi and Panda (2011)[2]. The orders L_u and L_y for the regressors were set to their largest delays presented in the second column of Tab. 7.

We compare the performances of the following GP models: standard GP-NARX, GP with Student-t likelihood and VB inference (GP-tVB) and GP with Laplace likelihood and EP inference (GP-LEP). Note that in order to use both GP-tVB and GP-LEP for system identification we simply need to build the appropriate regressors $\boldsymbol{x}_i = [y_{i-1},\cdots,y_{i-L_y},u_{i-1},\cdots,u_{i-L_u}]^\top$ and apply them as model inputs, following standard GP-NARX training and testing. The obtained root mean square errors (RMSE) for free simulation on test data are presented in Tab. 8.

In almost all scenarios with outliers both robust variants presented better performances than GP-NARX. Only in one case, *Artificial 3* dataset with 10% of corruption, GP-NARX performed better than one of the robust models (GP-tVB). In the scenarios without outliers, i.e., with Gaussian noise only, the GP-NARX model achieved the best RMSE for *Artificial 1* and *4* datasets, but it also performed close to the robust models for

[2]We opted here to follow Majhi and Panda (2011) and sample the outliers from an uniform distribution instead of, e.g., from a Student-t or Laplace distributions, because this could favor GP-tVB or GP-LEP, respectively. We note however that in the next sections of this chapter we adopt a different contamination strategy, explained later.

Table 8 – Summary of test free simulation RMSE values in scenarios with different contamination rates by outliers in the estimation data.

	Artificial 1				**Artificial 2**			
% of outliers	0%	2.5%	5%	10%	0%	2.5%	5%	10%
GP-NARX	**0.2134**	0.3499	0.3874	0.4877	0.3312	0.3724	0.5266	0.4410
GP-tVB	0.2455	0.3037	0.2995	**0.2868**	**0.3189**	**0.3247**	**0.3284**	**0.3306**
GP-LEP	0.2453	**0.2724**	**0.2720**	0.3101	0.3450	0.3352	0.3471	0.3963
	Artificial 3				**Artificial 4**			
GP-NARX	0.1106	0.4411	0.7022	0.6032	**0.6384**	2.1584	2.2935	2.4640
GP-tVB	0.1097	**0.1040**	**0.3344**	0.8691	0.6402	**0.7462**	2.2220	**2.1951**
GP-LEP	**0.0825**	0.3527	0.4481	**0.5738**	0.9188	1.1297	**2.1742**	2.3762
	Artificial 5							
GP-NARX	0.0256	0.0751	0.1479	0.1578				
GP-tVB	**0.0216**	0.0542	**0.0568**	**0.1006**				
GP-LEP	0.0345	**0.0499**	0.0747	0.1222				

the other datasets with 0% of corruption.

A good resilience to outliers was obtained for *Artificial 1* and *2* datasets, with GP-LEP and GP-tVB models being less affected in the cases with outliers. The most impressive performance was the one achieved by the GP-tVB model for all cases of the *Artificial 2* dataset, with little RMSE degradation.

For the *Artificial 3* dataset, only the GP-tVB model with 2.5% of outliers achieved error values close to the scenario without outliers. In the other cases, both variants, although better than standard GP-NARX model, presented considerably greater RMSE values than their results for 0% of outliers.

Likewise, in the experiments with *Artificial 4* and *5* datasets, we also observed that all models were affected by the corruption of the estimation data, even with lower quantities of outliers. However, it is important to emphasize that both GP-tVB and GP-LEP models achieved better RMSE values than GP-NARX, often by a large margin, as observed in the *Artificial 4* dataset for the GP-tVB model. In such cases, the robust variants can be considered a valid improvement over the standard GP-NARX model.

Finally, it is worth mentioning that during the experiments, the variational approach of the GP-tVB model has been consistently more stable than the EP algorithm of the GP-LEP model, even with the incorporation of the numerical safeties suggested by Rasmussen and Williams (2006) and Kuss (2006), which might be a decisive factor when choosing which model to apply for system identification.

Although the robust variants have performed better in the scenarios with outliers, we cannot state categorically that they were insensitive to the corrupted data, for both GP-tVB and GP-LEP models obtained noticeably worse RMSE in some cases with outliers. Depending on the task in hand, such degradation may or may not be tolerable. This observation, as well as some numerical issues encountered in the EP algorithm, encouraged us to further pursue alternative GP-based models which are more appropriate for robust system identification. In that context, we introduce in the next sections two new GP-based formulations specifically designed to tackle dynamical data contaminated by non-Gaussian noise.

4.2 GP-RLARX: Robust GP Latent Autoregressive Model

Both GP-tVB and GP-LEP models, presented in the previous section, handle the observed outliers by choosing a heavy-tailed likelihood. However, as we have previously argued, the feedback of noisy output values (possibly contaminated by outliers) in an autoregressive context can further compromise the learning procedure with dynamical data.

To this extent, we propose a robust alternative to the GP-NARX model that, besides incorporating the heavy-tailed Student-t distribution, also introduces a latent autoregressive structure to avoid outliers being directly used as input regressors.

This new robust GP model with latent autoregressive structure is named henceforth GP-RLARX and is defined by the following equations:

$$x_i = f(\bar{\boldsymbol{x}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}) + \varepsilon_i^{(x)}, \qquad \boldsymbol{f} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_f\right), \quad \varepsilon_i^{(x)} \sim \mathcal{N}\left(0, \sigma_x^2\right), \tag{4.17}$$

$$y_i = x_i + \varepsilon_i^{(y)}, \qquad \varepsilon_i^{(y)} \sim \mathcal{N}\left(0, \tau_i^{-1}\right), \quad \tau_i \sim \Gamma(\alpha, \beta), \tag{4.18}$$

where $\bar{\boldsymbol{x}}_{i-1} = [x_{i-1}, \cdots, x_{i-L}]^\top$ and $\bar{\boldsymbol{u}}_{i-1} = [u_{i-1}, \cdots, u_{i-L_u}]^\top$ are, respectively, autoregressive vectors of dynamical latent variables $x_i \in \mathbb{R}$ and external inputs $u_i \in \mathbb{R}$, L is the number of considered past latent variables and we have followed the same notation used in the previous sections. The unknown function $f(\cdot)$ has a GP prior with covariance matrix $\boldsymbol{K}_f$ and we have defined $\varepsilon_i^{(x)} \in \mathbb{R}$ as a zero mean Gaussian transition noise with variance σ_x^2. The observation noise $\varepsilon_i^{(y)} \in \mathbb{R}$ follows a Student-t distribution, which is written using the scale-mixture representation, i.e., it is considered to be sampled from a Gaussian whose precision τ_i has a gamma prior with hyperparameters α and β. Note that the GP-RLARX

model formulation presents a similar latent autoregressive structure of the RGP model[3], introduced in Chapter 3, Section 3.2.

We emphasize that Eq. (4.17) is distinct from standard GP-NARX Eqs. (3.3) and (3.4) presented in Section 3.1.1, which are also used by the GP-tVB and GP-LEP models. In the GP-RLARX formulation, the autoregression is made with the dynamical latent variables x_i, instead of the observed outputs y_i. This feature avoids the feedback of possibly corrupted observations into the dynamics. Furthermore, differently from the inputs of standard NARX models, the latent variables x_i have a probability distribution, which enables the propagation of uncertainty during free simulation.

The features proposed for the GP-RLARX model make it more powerful, but also introduce additional intractabilities not covered, for instance, by the variational framework of GP-tVB. These additional intractabilities come from the difficulty in propagating the uncertainty of latent inputs through the nonlinear GP prior. In order to overcome this issue, we build on the variational approach of the Bayesian GP-LVM (TITSIAS; LAWRENCE, 2010) and extend it to account for the Student-t likelihood of the GP-RLARX and its latent autoregressive structure, the latter similarly handled by the REVARB method presented in Section 3.3 for the RGP model.

We start by rewriting Eqs. (4.17) and (4.18) in terms of distributions:

$$
\begin{aligned}
p(\boldsymbol{f}|\hat{\boldsymbol{X}}) &= \mathcal{N}(\boldsymbol{f}|\boldsymbol{0},\boldsymbol{K}_f), & \\
p(x_i) &= \mathcal{N}(x_i|\mu_{0i},\lambda_{0i}), & 1 \leq i \leq L, \\
p(x_i|f_i) &= \mathcal{N}(x_i|f_i,\sigma_x^2), & L+1 \leq i \leq N, \\
p(y_i|x_i,\tau_i) &= \mathcal{N}(y_i|x_i,\tau_i^{-1}), & L+1 \leq i \leq N, \\
p(\tau_i) &= \Gamma(\tau_i|\alpha,\beta), & L+1 \leq i \leq N,
\end{aligned}
$$

where $\hat{\boldsymbol{X}} \in \mathbb{R}^{(N-L)\times(L+L_u)}$ is the stack of input vectors $\hat{\boldsymbol{x}}_i = [\bar{\boldsymbol{x}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}]^\top \in \mathbb{R}^{L+L_u}, L+1 \leq i \leq N$, $\boldsymbol{K}_f \in \mathbb{R}^{(N-L)\times(N-L)}$ is the covariance matrix of the GP and we have put Gaussian priors with moments μ_{0i} and λ_{0i} to the initial latent variables $x_i|_{i=1}^{L}$.

We follow a variational procedure similar to the REVARB framework derivation presented in Section 3.3, keeping the same notation. Thus, we include M inducing points $\boldsymbol{z} \in \mathbb{R}^M$ evaluated in M pseudo-inputs $\boldsymbol{\zeta}_j|_{j=1}^{M} \in \mathbb{R}^{L+L_u}$, where $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{0},\boldsymbol{K}_z)$

[3]Chronologically speaking, work on the GP-RLARX model was actually developed and published by us before the RGP model.

and $\boldsymbol{K}_z \in \mathbb{R}^{M\times M}$ is the covariance matrix computed from the pseudo-inputs. The joint distribution of all the variables is now given by

$$p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{f},\boldsymbol{z},\boldsymbol{\tau}) = \left(\prod_{i=L+1}^{N} p(y_i|x_i,\tau_i)p(\tau_i)p(x_i|f_i)p(f_i|\boldsymbol{z},\hat{\boldsymbol{x}}_i)\right) p(\boldsymbol{z})\prod_{i=1}^{L} p(x_i). \tag{4.19}$$

Note that if we integrate out $\boldsymbol{z}$ we recover exactly the original model without inducing points.

Applying Jensen's inequality to Eq. (4.19) gives a lower bound to the marginal log-likelihood $\log p(\boldsymbol{y})$:

$$\log p(\boldsymbol{y}) \geq \int_{\boldsymbol{x},\boldsymbol{f},\boldsymbol{z},\boldsymbol{\tau}} Q \log\left[\frac{p(\boldsymbol{y},\boldsymbol{x},\boldsymbol{f},\boldsymbol{z},\boldsymbol{\tau})}{Q}\right], \tag{4.20}$$

where Q is the variational distribution, chosen to be given as follows:

$$Q = q(\boldsymbol{x})q(\boldsymbol{z})q(\boldsymbol{\tau})\prod_{i=L+1}^{N} p(f_i|\boldsymbol{z},\hat{\boldsymbol{x}}_i). \tag{4.21}$$

Each term of the variational posterior is defined by

$$q(\boldsymbol{x}) = \prod_{i=1}^{N} q(x_i) = \prod_{i=1}^{N} \mathcal{N}(x_i|\mu_i,\lambda_i), \tag{4.22}$$

$$q(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{m},\boldsymbol{S}), \tag{4.23}$$

$$q(\boldsymbol{\tau}) = \prod_{i=L+1}^{N} q(\tau_i) = \prod_{i=L+1}^{N} \Gamma(\tau_i|a_i,b_i), \tag{4.24}$$

$$p(f_i|\boldsymbol{z},\hat{\boldsymbol{x}}_i) = \mathcal{N}(f_i|[\boldsymbol{a}_f]_i,[\boldsymbol{\Sigma}_f]_{ii}), \tag{4.25}$$

where $\boldsymbol{a}_f = \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{z}$,

$$\boldsymbol{\Sigma}_f = \boldsymbol{K}_f - \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{K}_{fz}^{\top}.$$

The variables $\mu_i \in \mathbb{R}$, $\lambda_i \in \mathbb{R}_{>0}$, $\boldsymbol{m} \in \mathbb{R}^M$, $\boldsymbol{S} \in \mathbb{R}^{M\times M}$, $a_i, b_i \in \mathbb{R}_{>0}$ are variational parameters and $\boldsymbol{K}_{fz} \in \mathbb{R}^{(N-L)\times M}$ is the cross-covariance matrix calculated from $\hat{\boldsymbol{x}}_i|_{i=L+1}^{N}$ and $\boldsymbol{\zeta}_j|_{j=1}^{M}$.

We replace the factorized variational distribution Q back to Eq. (4.20) and by working the expressions, with the same strategy followed by the REVARB steps presented in the Appendix A.2, we are able to optimally eliminate the variational parameters $\boldsymbol{m}$ and $\boldsymbol{S}$. After the derivation, detailed in the Appendix A.3, we obtain a lower bound to the

marginal log-likelihood $\log p(\boldsymbol{y})$ of the GP-RLARX model:

$$\begin{aligned}\log p(\boldsymbol{y}) \geq & -\frac{N-L}{2}\left(\log 2\pi\sigma_x^2 - \log 2\pi\right) + \frac{1}{2}\sum_{i=L+1}^{N}\left(\psi(a_i) - \log b_i\right) \\ & - \frac{1}{2}\sum_{i=L+1}^{N}\left[\frac{a_i}{b_i}\left(y_i^2 - 2y_i\mu_i + \lambda_i + \mu_i^2\right)\right] - \frac{1}{2\sigma_x^2}\left[\sum_{i=1}^{N}\left(\lambda_i + \mu_i^2\right) + \Psi_0 - \mathrm{Tr}(\boldsymbol{K}_z^{-1}\boldsymbol{\Psi}_2)\right] \\ & + \frac{1}{2}\log|\boldsymbol{K}_z| - \frac{1}{2}\log\left|\boldsymbol{K}_z + \frac{1}{\sigma_x^2}\boldsymbol{\Psi}_2\right| + \frac{1}{2(\sigma_x^2)^2}\boldsymbol{\mu}^\top\boldsymbol{\Psi}_1\left(\boldsymbol{K}_z + \frac{1}{\sigma_x^2}\boldsymbol{\Psi}_2\right)^{-1}\boldsymbol{\Psi}_1^\top\boldsymbol{\mu} \\ & - \sum_{i=1}^{N}\int_{x_i} q(x_i)\log q(x_i) + \sum_{i=1}^{L}\int_{x_i} q(x_i)\log p(x_i) - \mathrm{KL}(q(\boldsymbol{\tau})\,\|\,p(\boldsymbol{\tau})),\end{aligned} \tag{4.26}$$

where $\psi(a_i) = \frac{\partial \log\Gamma(a_i)}{\partial a_i}$ is the digamma function and the last term is the Kullback-Leibler divergence between two gamma distributions. We have once more used the statistics $\Psi_0 = \mathrm{Tr}(\langle \boldsymbol{K}_f\rangle_{q(\boldsymbol{x})})$, $\boldsymbol{\Psi}_1 = \langle \boldsymbol{K}_{fz}\rangle_{q(\boldsymbol{x})}$ and $\boldsymbol{\Psi}_2 = \langle \boldsymbol{K}_{fz}^\top \boldsymbol{K}_{fz}\rangle_{q(\boldsymbol{x})}$, where $\langle\cdot\rangle_{q(\boldsymbol{x})}$ denotes expectation with respect to the distribution $q(\boldsymbol{x})$. Those expressions, which are tractable for the squared exponential kernel, are identical to the ones presented in the Appendix A.1 for the REVARB method.

We can also write a compact version for the GP-RLARX lower bound as follows:

$$\log p(\boldsymbol{y}) \geq \sum_{i=L+1}^{N}\left[\mathscr{L}_i^{(y)} + \mathscr{L}_i^{(x)}\right] + \sum_{i=1}^{N}\mathscr{H}_i + \sum_{i=1}^{L}\mathscr{L}_{0i} - \mathrm{KL}\left(q\left(\boldsymbol{z}\right)||p\left(\boldsymbol{z}\right)\right) - \mathrm{KL}\left(q\left(\boldsymbol{\tau}\right)||p\left(\boldsymbol{\tau}\right)\right), \tag{4.27}$$

where each term is given by:

$$\mathscr{L}_i^{(y)} = \left\langle \log p\left(y_i|x_i,\tau_i\right)\right\rangle_{q(\boldsymbol{x})q(\boldsymbol{\tau})}, \tag{4.28}$$

$$\mathscr{L}_i^{(x)} = \left\langle p\left(f_i|\boldsymbol{z},\hat{\boldsymbol{x}}_i\right)\log p\left(x_i|f_i\right)\right\rangle_{q(\boldsymbol{x})q(\boldsymbol{z})}, \tag{4.29}$$

$$\mathscr{H}_i = -\left\langle \log q\left(x_i\right)\right\rangle_{q(\boldsymbol{x})}, \tag{4.30}$$

$$\mathscr{L}_{0i} = \left\langle \log p\left(x_i\right)\right\rangle_{q(\boldsymbol{x})}, \tag{4.31}$$

$$\mathrm{KL}\left(q\left(\boldsymbol{z}\right)||p\left(\boldsymbol{z}\right)\right) = \int_{\boldsymbol{z}} q\left(\boldsymbol{z}\right)\log q\left(\boldsymbol{z}\right) - \int_{\boldsymbol{z}} q\left(\boldsymbol{z}\right)\log p\left(\boldsymbol{z}\right), \tag{4.32}$$

$$\mathrm{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau})) = \sum_{i=L+1}^{N}\left[(a_i-\alpha)\psi(a_i) + \log\frac{\Gamma(\alpha)}{\Gamma(a_i)} + \alpha\log\frac{b_i}{\beta} + a_i\frac{\beta-b_i}{b_i}\right]. \tag{4.33}$$

It is worth mentioning that the second line of the full GP-RLARX bound in Eq. (4.26) shows the only term that includes the observations. The value of each observation

y_i is weighted by the fraction a_i/b_i, which comes from the expectation with respect to the variational distribution $q(\boldsymbol{\tau})$ in Eq. (4.28). For observations containing outliers, the value of this fraction is much lower than regular observations, reducing their influence in the bound. Thus, the inspection of the optimized variational parameters a_i and b_i *per se* can be used as a method to detect outliers in the estimation data. It is also important to notice that the full GP-RLARX bound in Eq. (4.26) is not factorized along the observations.

The kernel hyperparameters and the variational parameters, including the pseudo-inputs $\boldsymbol{\zeta}_j|_{j=1}^{M}$, are jointly optimized by maximizing the GP-RLARX lower bound, using the analytical gradients of Eq. (4.26), which constitutes its model selection step. Such optimization is performed by standard gradient-based methods, such as the BFGS algorithm (FLETCHER, 2013).

Although the computation of the exact predictive distribution for the GP-RLARX model is intractable, since the input is uncertain, we can calculate its moments, similar to the REVARB framework in Section 3.3.1. Given a new regressor vector $\hat{\boldsymbol{x}}_*$ obtained from past latent variables and a sequence of external inputs, the mean and variance of the correspondent prediction f_* are calculated by following the results presented by Girard *et al.* (2002), Girard *et al.* (2003) using moment matching:

$$p(f_*) = \langle p(f_*|\hat{\boldsymbol{x}}_*)\rangle_{q(\boldsymbol{x}_*)} \approx \mathcal{N}(f_*|\mu_*, \lambda_*), \tag{4.34}$$

$$\mu_* = \boldsymbol{B}^\top(\boldsymbol{\Psi}_1^*)^\top, \tag{4.35}$$

$$\lambda_* = \boldsymbol{B}^\top(\boldsymbol{\Psi}_2^* - (\boldsymbol{\Psi}_1^*)^\top\boldsymbol{\Psi}_1^*)\boldsymbol{B} + \Psi_0^* - \mathrm{Tr}((\boldsymbol{K}_z^{-1} - (\boldsymbol{K}_z + \sigma_x^{-2}\boldsymbol{\Psi}_2)^{-1})\boldsymbol{\Psi}_2^*), \tag{4.36}$$

where $\boldsymbol{B} = \sigma_x^{-2}(\boldsymbol{K}_z + \sigma_x^{-2}\boldsymbol{\Psi}_2)^{-1}\boldsymbol{\Psi}_1^\top\boldsymbol{\mu}$ and the statistics Ψ_0^*, $\boldsymbol{\Psi}_1^*$ and $\boldsymbol{\Psi}_2^*$ are computed as before, in the lower bound, but using the new approximation $q(x_*) = \mathcal{N}\left(x_*\middle|\mu_*, \lambda_* + \sigma_x^2\right)$. In the output, the predicted variance λ_* can be added by the median[4] of the fractions $b_i/a_i|_{i=L+1}^{N}$ related to the training data, since the variance of the Student-t likelihood is β/α (see Eq. (4.5)). Thus, the moments of the final prediction are given by $\mathbb{E}\{y_*\} = \mu_*$ and $\mathbb{V}\{y_*\} = \lambda_* + \mathrm{median}(b_i/a_i|_{i=L+1}^{N})$.

Importantly, predictions are not made directly with the observations $\boldsymbol{y}$, since they could contain outliers, but with the variational moments of the dynamical latent variables, used to compute the matrix $\boldsymbol{B}$. Furthermore, the optimized variational means $\boldsymbol{\mu}$, as we will illustrate later, act as a filtered version of the observed outputs. As opposed

[4] In this case it is better to take the median value instead of the mean, since the fractions related to the outliers can be much larger.

Algorithm 3: GP-RLARX for outlier-robust dynamical modeling.

- Estimation step

Require: $\boldsymbol{u} \in \mathbb{R}^N$ (external input), $\boldsymbol{y} \in \mathbb{R}^N$ (output), M (number of inducing points), L (latent order lag), L_u (input order lag)

Initialize kernel hyperparameters and variational parameters;

repeat

Compute the evidence lower bound with Eq. (4.26);

Compute the analytical gradients of Eq. (4.26) with respect to the unknown parameters;

Update parameters with a gradient-based method (e.g. BFGS);

until convergence or maximum number or iterations

Output the optimized parameters;

Check the ratios $a_i/b_i|_{i=L+1}^{N}$, the smallest values are related to the estimation samples which probably contain outliers;

- Free simulation with test data

Require: Test external inputs $\boldsymbol{u}_* \in \mathbb{R}^{N_*}$ and the previously estimated GP-RLARX model

for $i = 1 : N_*$ **do**

Compute the predictive mean μ_{*i} and variance λ_{*i} with Eqs. (4.35) and (4.36);

Update the variational distribution of the new latent dynamical variable with $q(x_{*i}) = \mathcal{N}\left(x_{*i} \middle| \mu_{*i}, \lambda_{*i} + \sigma_x^2\right)$;

Output $y_{*i} \sim \mathcal{N}\left(\mu_{*i}, \lambda_{*i} + \text{median}(b_i/a_i|_{i=L+1}^{N})\right)$;

end for

to GP-tVB and other standard NARX models, the GP-RLARX framework allows for a natural way of approximate propagating the uncertainty during free simulation, since it recursively uses the full predictive distributions as inputs for next predictions, instead of just their mean values or single point estimates.

The GP-RLARX model was firstly introduced by us in Mattos *et al.* (2016), where it performed well compared to GP-NARX and GP-tVB in several artificial benchmarks with different levels of outlier contamination. We will postpone the reproduction of those experiments for now, since in the next section we will present a robust version of the multilayer RGP model described in Section 3.2, which constitutes a step further from GP-RLARX in both model complexity (and computational requirements) but hopefully also in model expressiveness.

4.3 The RGP-*t* Model

Despite the superior performance by the GP-RLARX model in the presence of outliers when compared to GP-tVB, as reported in Mattos *et al.* (2016), its framework adopts a very simple observation model: the output y_i is equal to the current latent variable

x_i plus the noise $\boldsymbol{\varepsilon}_i^{(y)}$ (see Eq. (4.18)). This forces the latent space to be closely related to the output and constrains the latent variables. Furthermore, it relies on a single transition layer, which does not support any kind of hierarchy.

The aforementioned restrictions could hinder the GP-RLARX model's capability to learn more complex dynamics from outlier-corrupted data. Bearing this in mind, we can extend GP-RLARX by including an additional GP to model the observation (or emission) layer, in order to separate the transition and emission nonlinear functions. To further increase the representational capability of the model, we allow the inclusion of more than one transition layer, referred to as *hidden layers*. This approach builds upon the RGP model, described in Section 3.2, being equivalent to provide it with a robust Student-t likelihood. However, such extension requires a modification to the REVARB framework previously presented in Section 3.3, since it originally considered a Gaussian likelihood, which makes it not suitable to scenarios where outliers are expected to occur.

The extension proposed by us in Mattos *et al.* (2017) aims to bring together in a synergistic way the best properties of the GP-RLARX and RGP models in order to eventually obtain a more reliable solution for robust system identification: (*i*) *resilience to non-Gaussian noise* provided by the GP-RLARX model due to the Student-t likelihood, and (*ii*) the enhanced *representational capability* provided by the hierarchical RGP structure. We name henceforth this new model as the RGP-t model and the corresponding modified variational framework as REVARB-t.

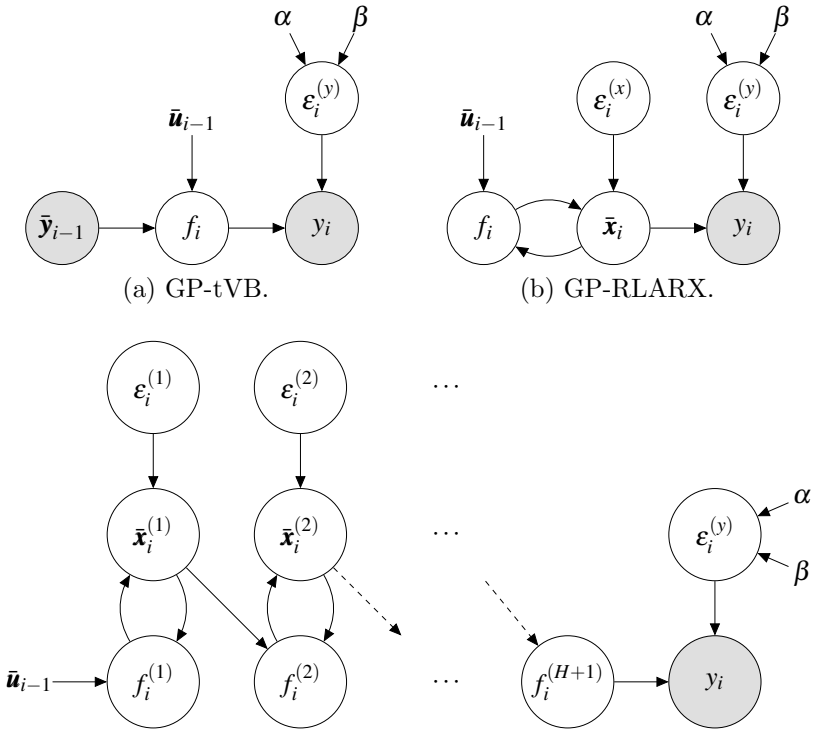
Considering H hidden transition layers and one observation layer, the RGP-t model is characterized by the following set of equations:

$$x_i^{(h)} = f^{(h)}\left(\hat{\boldsymbol{x}}_i^{(h)}\right) + \varepsilon_i^{(h)}, \qquad \varepsilon_i^{(h)} \sim \mathcal{N}\left(0, \sigma_h^2\right), \quad 1 \leq h \leq H, \tag{4.37}$$

$$y_i = f^{(H+1)}\left(\hat{\boldsymbol{x}}_i^{(H+1)}\right) + \varepsilon_i^{(H+1)}, \qquad \varepsilon_i^{(H+1)} \sim \mathcal{N}\left(0, \tau_i^{-1}\right), \quad \tau_i \sim \Gamma(\alpha, \beta), \tag{4.38}$$

$$\boldsymbol{f}^{(h)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_f^{(h)}\right), \qquad 1 \leq h \leq H+1. \tag{4.39}$$

Each layer is related to an unknown function $f^{(h)}(\cdot)$ modeled by a distinct GP with covariance matrix $\boldsymbol{K}_f^{(h)}$. We emphasize that, while the transition noises $\boldsymbol{\varepsilon}_i^{(h)}$ are Gaussian, the observation noise $\boldsymbol{\varepsilon}_i^{(H+1)}$ is Student-t, written in the scale-mixture representation (Gaussian with gamma distributed precision), and capable of handling outliers. Differently from the GP-RLARX model, the RGP-t models at least two nonlinear functions, when $H = 1$, which corresponds to at least two separate GP priors.



(a) GP-tVB. (b) GP-RLARX.

(c) RGP-t with H hidden layers.

Figure 26 – Graphical models for some of the robust GP-based approaches considered for dynamical modeling in this chapter. The shaded nodes are random observed entities (outputs), while the white nodes represent the latent (unobserved) variables. The arrows indicate direct dependencies. Note that past exogenous inputs $\bar{\boldsymbol{u}}_{i-1}$ are treated as deterministic elements.

Similar to the original RGP, in the previous equations, each layer's input $\hat{\boldsymbol{x}}_i^{(h)} \in \mathbb{R}^{D_h}$ is a regressor vector given by Eq. (3.14), repeated here for convenience:

$$\hat{\boldsymbol{x}}_i^{(h)} = \begin{cases} \left[\bar{\boldsymbol{x}}_{i-1}^{(1)}, \bar{\boldsymbol{u}}_{i-1}\right]^\top, & \text{if } h = 1, \\ \left[\bar{\boldsymbol{x}}_{i-1}^{(h)}, \bar{\boldsymbol{x}}_i^{(h-1)}\right]^\top, & \text{if } 1 < h \leq H, \\ \bar{\boldsymbol{x}}_i^{(H)}, & \text{if } h = H+1, \end{cases} \tag{4.40}$$

where $\bar{\boldsymbol{x}}_{i-1}^{(h)} = \left[x_{i-1}^{(h)}, \cdots, x_{i-L}^{(h)}\right]$, and $\bar{\boldsymbol{u}}_{i-1} = [u_{i-1}, \cdots, u_{i-L_u}]$.

Importantly, all the model dynamics learned from the latent autoregressive states $\bar{\boldsymbol{x}}_{i-1}^{(h)}$ presented in Eq. (4.40) do not depend directly from the possibly outlier-corrupted observations.

Fig. 26 illustrates the structures of some of the different outlier-robust GP-based models used in this chapter for system identification: the GP-tVB, the GP-RLARX and the RGP-t. In the graphical models, which follow the notation used so far, the shaded nodes are random observed entities (outputs), the white nodes represent the latent (unobserved) variables and the past exogenous inputs $\bar{\boldsymbol{u}}_{i-1}$ are treated as deterministic

elements. The arrows indicate direct dependencies, e.g., the observation y_i depends on the noise $\varepsilon_i^{(y)}$. We opted to make explicit the noise models and the latent f_i variables, even though the latter are always integrated out (i.e., marginalized). It should be noted that the models include the hyperparameters α and β related to the gamma distributed precision of the observation noise. The variance hyperparameter of the Gaussian transition noises are not shown, but are implicit. The recent robust model presented by Ranjan *et al.* (2016), mentioned in the beginning of this chapter, follows the same NARX structure of the GP-tVB, although with a different inference method (an EM algorithm).

The original RGP model is trained with the REVARB framework, proposed by us in Mattos *et al.* (2016) and described in Section 3.3. Since the standard REVARB considers only a Gaussian likelihood, we now present a robust modification in order to train the RGP-t model, named REVARB-t, introduced by us in Mattos *et al.* (2017) and explained in the next section.

Remark The model formulation evolution from the Student-t equipped GP-tVB to the GP-RLARX, and then to the RGP-t, represents a pursuit towards more expressive models that are, at the same time, robust to outliers and tailored to nonlinear system identification. As can be inferred from Fig. 26, the GP-tVB is simply an adaptation of a robust regression model to the NARX structure. The GP-RLARX then introduces latent dynamical variables, which, although closely related to the observations, are properly handled as random variables. Finally, the RGP-t enhances the previous approaches by incorporating a fully hierarchical structure, increasing its representational power, while also preserving the focus on simulation with uncertainty propagation through its dynamical transitions and a robust nonlinear observation layer.

4.4 The REVARB-t Framework

For the sake of clarity, we follow here the steps of the original REVARB method described in Chapter 3, Section 3.3, including elements of the GP-RLARX variational framework (Section 4.2) and emphasizing the modifications that led to the proposal of REVARB-t. Though possibly redundant at some points, such approach will make this section more readable.

We first rewrite Eqs. (4.37) and (4.38) to explicit the involved distributions:

$$
\begin{aligned}
p\left(\boldsymbol{f}^{(h)}\middle|\hat{\boldsymbol{X}}^{(h)}\right) &= \mathcal{N}\left(\boldsymbol{f}^{(h)}\middle|\boldsymbol{0},\boldsymbol{K}_f^{(h)}\right), && 1 \leq h \leq H+1,\\
p\left(x_i^{(h)}\right) &= \mathcal{N}\left(x_i^{(h)}\middle|\mu_{0i}^{(h)},\lambda_{0i}^{(h)}\right), && 1 \leq i \leq L,\\
p\left(x_i^{(h)}\middle|f_i^{(h)}\right) &= \mathcal{N}\left(x_i^{(h)}\middle|f_i^{(h)},\sigma_h^2\right), && L+1 \leq i \leq N,\\
p\left(y_i\middle|f_i^{(H+1)},\tau_i\right) &= \mathcal{N}\left(y_i\middle|f_i^{(H+1)},\tau_i^{-1}\right), && L+1 \leq i \leq N,\\
p(\tau_i) &= \Gamma(\tau_i|\alpha,\beta), && L+1 \leq i \leq N,
\end{aligned}
$$

where $\hat{\boldsymbol{X}}^{(h)}$ is the stack of input vectors $\hat{\boldsymbol{x}}_i^{(h)}|_{i=L+1}^{N}$, and the means $\mu_{0i}^{(h)} \in \mathbb{R}$ and variances $\lambda_{0i}^{(h)} \in \mathbb{R}_{>0}$ come from the Gaussian priors in the initial L latent variables of each layer. Interestingly, if we sample from such generative model, the presence of the gamma prior $p(\tau_i) = \Gamma(\tau_i|\alpha,\beta)$ would enable the generation of a few small precisions τ_i (high variances) for the observation noise values, differently from the original RGP, which allows only Gaussian distributed observation noise.

One more time we tackle the model's intractabilities by applying Titsias' sparse variational framework (TITSIAS, 2009a). We augment each layer h by including M inducing points $\boldsymbol{z}^{(h)} \in \mathbb{R}^M$ evaluated in M pseudo-inputs $\boldsymbol{\zeta}_j^{(h)}|_{j=1}^{M} \in \mathbb{R}^{D_h}$, where D_h is the same dimension of $\hat{\boldsymbol{x}}_i^{(h)}$ and $p\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)}\middle|\boldsymbol{0},\boldsymbol{K}_z^{(h)}\right)$, where $\boldsymbol{K}_z^{(h)} \in \mathbb{R}^{M\times M}$ is the covariance matrix obtained from $\boldsymbol{\zeta}_j^{(h)}|_{j=1}^{M}$. The joint distribution of the variables present in the RGP-t model is now given by

$$
\begin{aligned}
&p\left(\boldsymbol{y},\boldsymbol{\tau},\left\{\boldsymbol{x}^{(h)}\right\}\middle|_{h=1}^{H},\left\{\boldsymbol{f}^{(h)},\boldsymbol{z}^{(h)}\right\}\middle|_{h=1}^{H+1}\right) =\\
&\prod_{i=L+1}^{N} p\left(y_i\middle|f_i^{(H+1)},\tau_i\right) p(\tau_i) p\left(f_i^{(H+1)}\middle|\boldsymbol{z}^{(H+1)},\hat{\boldsymbol{x}}_i^{(H)}\right)\\
&\left(\prod_{h=1}^{H} p\left(x_i^{(h)}\middle|f_i^{(h)}\right) p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\right)\left(\prod_{h=1}^{H+1} p\left(\boldsymbol{z}^{(h)}\right)\right)\left(\prod_{i=1}^{L}\prod_{h=1}^{H} p\left(x_i^{(h)}\right)\right),
\end{aligned}
\tag{4.41}
$$

where the boldface indexless notation $\boldsymbol{x}^{(h)}$ refers to all the variables $x_i^{(h)}\big|_{i=1}^{N}$ within the h-th layer.

By applying Jensen's inequality we are able to obtain a lower bound to the marginal log-likelihood $\log p(\boldsymbol{y})$:

$$
\log p(\boldsymbol{y}) \geq \int_{\boldsymbol{\tau},\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} Q \log\left[\frac{p\left(\boldsymbol{y},\boldsymbol{\tau},\left\{\boldsymbol{x}^{(h)}\right\}\middle|_{h=1}^{H},\left\{\boldsymbol{f}^{(h)},\boldsymbol{z}^{(h)}\right\}\middle|_{h=1}^{H+1}\right)}{Q}\right],
\tag{4.42}
$$

where Q is the variational distribution. We conveniently choose the following factorized expression for Q:

$$Q = q(\boldsymbol{\tau}) \left(\prod_{h=1}^{H} q\left(\boldsymbol{x}^{(h)}\right) \right) \left(\prod_{h=1}^{H+1} q\left(\boldsymbol{z}^{(h)}\right) \right) \left(\prod_{i=L+1}^{N} \prod_{h=1}^{H+1} p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right) \right), \tag{4.43}$$

where $q(\boldsymbol{\tau})$, $q\left(\boldsymbol{x}^{(h)}\right)$ and $q\left(\boldsymbol{z}^{(h)}\right)$ are respectively the variational posterior distributions related to the latent precisions $\boldsymbol{\tau}$, the latent dynamical variables $\boldsymbol{x}^{(h)}$ and the inducing points $\boldsymbol{z}^{(h)}$.

Considering a mean-field approximation, each term is given by

$$q(\boldsymbol{\tau}) = \prod_{i=L+1}^{N} \Gamma(\tau_i | a_i, b_i), \tag{4.44}$$

$$q\left(\boldsymbol{x}^{(h)}\right) = \prod_{i=1}^{N} \mathcal{N}\left(x_i^{(h)} \middle| \mu_i^{(h)}, \lambda_i^{(h)}\right), \tag{4.45}$$

$$q\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{m}^{(h)}, \boldsymbol{S}^{(h)}\right), \tag{4.46}$$

$$p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right) = \mathcal{N}\left(f_i^{(h)} \middle| \left[\boldsymbol{a}_f^{(h)}\right]_i, \left[\boldsymbol{\Sigma}_f^{(h)}\right]_{ii}\right), \tag{4.47}$$

where $\boldsymbol{a}_f^{(h)} = \boldsymbol{K}_{fz}^{(h)} \left(\boldsymbol{K}_z^{(h)}\right)^{-1} \boldsymbol{z}^{(h)}$ and $\boldsymbol{\Sigma}_f^{(h)} = \boldsymbol{K}_f^{(h)} - \boldsymbol{K}_{fz}^{(h)} \left(\boldsymbol{K}_z^{(h)}\right)^{-1} \left(\boldsymbol{K}_{fz}^{(h)}\right)^{\top}$.

In the above, $a_i, b_i \in \mathbb{R}_{>0}$, $\mu_i^{(h)} \in \mathbb{R}$, $\lambda_i^{(h)} \in \mathbb{R}_{>0}$, $\boldsymbol{m}^{(h)} \in \mathbb{R}^M$ and $\boldsymbol{S}^{(h)} \in \mathbb{R}^{M \times M}$ are variational parameters, $\boldsymbol{K}_f^{(h)} \in \mathbb{R}^{(N-L)\times(N-L)}$ is the standard kernel matrix obtained from $\hat{\boldsymbol{x}}_i^{(h)}|_{i=L+1}^{N}$, $\boldsymbol{K}_z^{(h)} \in \mathbb{R}^{M \times M}$ is the sparse kernel matrix calculated from the pseudo-inputs $\boldsymbol{\zeta}_j^{(h)}|_{j=1}^{M}$ and $\boldsymbol{K}_{fz}^{(h)} \in \mathbb{R}^{(N-L)\times M}$, where $\left[\boldsymbol{K}_{fz}^{(h)}\right]_{ij} = k\left(\hat{\boldsymbol{x}}_i^{(h)}, \boldsymbol{\zeta}_j^{(h)}\right)$. Note that while the hyperparameters of the gamma prior α and β are global, each sample is related to its own set of local variational posterior parameters a_i and b_i.

Replacing the factorized variational distribution Q back in the Eq. (4.42) we are able to cancel out the intractable terms $p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right)$ inside the logarithm. We are also able to optimally eliminate the variational parameters $\boldsymbol{m}^{(h)}$ and $\boldsymbol{S}^{(h)}$ following the REVARB steps in the Appendix A.2. After solving the tractable integrals and some algebraic manipulation detailed in the Appendix A.4, we finally get to the REVARB-t

lower bound:

$$
\begin{aligned}
\log p(\boldsymbol{y}) \geq & -\frac{N-L}{2}\sum_{h=1}^{H}\log 2\pi\sigma_h^2 - \frac{N-L}{2}\log 2\pi + \frac{1}{2}\sum_{i=L+1}^{N}\left(\psi(a_i)-\log b_i\right) \\
& -\frac{1}{2}\left(\boldsymbol{y}^\top \boldsymbol{R}\boldsymbol{y} + \Psi_0^{\prime(H+1)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{\prime(H+1)}\right)\right) \\
& +\frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)} + \boldsymbol{\Psi}_2^{\prime(H+1)}\right| \\
& +\frac{1}{2}\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{\prime(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \boldsymbol{\Psi}_2^{\prime(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{\prime(H+1)}\right)^\top \boldsymbol{y} - \mathrm{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau})) \\
& +\sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right. \\
& -\frac{1}{2\sigma_h^2}\Psi_0^{(h)} + \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right| \\
& +\frac{1}{2(\sigma_h^2)^2}\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} \\
& \left.-\sum_{i=1}^{N}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right)\right\},
\end{aligned}
\tag{4.48}
$$

where we have once more defined the statistics $\Psi_0^{(h)} = \mathrm{Tr}\left(\left\langle \boldsymbol{K}_f^{(h)}\right\rangle_{q(\boldsymbol{x}^{(h)})}\right)$, $\boldsymbol{\Psi}_1^{(h)} = \left\langle \boldsymbol{K}_{fz}^{(h)}\right\rangle_{q(\boldsymbol{x}^{(h)})}$ and $\boldsymbol{\Psi}_2^{(h)} = \left\langle \left(\boldsymbol{K}_{fz}^{(h)}\right)^\top \boldsymbol{K}_{fz}^{(h)}\right\rangle_{q(\boldsymbol{x}^{(h)})}$ for the hidden layers, i.e., $1 \leq h \leq H$. Those are identical to REVARB's expressions detailed in the Appendix A.1. For the output layer we have slightly different statistics given by

$$
\Psi_0^{\prime(H+1)} = \mathrm{Tr}\left(\boldsymbol{R}\left\langle \boldsymbol{K}_f^{(H+1)}\right\rangle_{q\left(\boldsymbol{x}^{(H)}\right)}\right), \tag{4.49}
$$

$$
\boldsymbol{\Psi}_1^{\prime(H+1)} = \boldsymbol{R}\left\langle \boldsymbol{K}_{fz}^{(H+1)}\right\rangle_{q\left(\boldsymbol{x}^{(H)}\right)}, \tag{4.50}
$$

$$
\boldsymbol{\Psi}_2^{\prime(H+1)} = \left\langle \left(\boldsymbol{K}_{fz}^{(H+1)}\right)^\top \boldsymbol{R}\boldsymbol{K}_{fz}^{(H+1)}\right\rangle_{q\left(\boldsymbol{x}^{(H)}\right)}, \tag{4.51}
$$

where $\boldsymbol{R} = \mathrm{diag}\left(\frac{a_{L+1}}{b_{L+1}}, \cdots, \frac{a_N}{b_N}\right)$.

Those new expressions, which are tractable when the exponentiated quadratic kernel is chosen, are also presented in the Appendix A.1.

The compact version of the REVARB-t bound is given by:

$$
\begin{aligned}
\log p(\boldsymbol{y}) \geq & \sum_{i=L+1}^{N}\sum_{h=1}^{H+1}\mathscr{L}_i^{(h)} + \sum_{i=1}^{N}\sum_{h=1}^{H}\mathscr{H}_i^{(h)} + \sum_{i=1}^{L}\sum_{h=1}^{H}\mathscr{L}_{0i}^{(h)} \\
& -\sum_{h=1}^{H+1}\mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right)\middle|\middle|p\left(\boldsymbol{z}^{(h)}\right)\right) - \mathrm{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau})),
\end{aligned}
\tag{4.52}
$$

where the terms are computed as below:

$$\mathscr{L}_i^{(H+1)} = \left\langle p\left(f_i^{(H+1)}\middle|\boldsymbol{z}^{(H+1)},\hat{\boldsymbol{x}}_i^{(H+1)}\right)\log p\left(y_i\middle|f_i^{(H+1)},\tau_i\right)\right\rangle_{q(\boldsymbol{x})q(\mathbf{z})q(\boldsymbol{\tau})}, \tag{4.53}$$

$$\mathscr{L}_i^{(h)} = \left\langle p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\log p\left(x_i^{(h)}\middle|f_i^{(h)}\right)\right\rangle_{q(\boldsymbol{x})q(\mathbf{z})}, \tag{4.54}$$

$$\mathscr{H}_i^{(h)} = -\left\langle \log q\left(x_i^{(h)}\right)\right\rangle_{q(\boldsymbol{x})}, \tag{4.55}$$

$$\mathscr{L}_{0i}^{(h)} = \left\langle \log p\left(x_i^{(h)}\right)\right\rangle_{q(\boldsymbol{x})}, \tag{4.56}$$

$$\mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right)\middle||p\left(\boldsymbol{z}^{(h)}\right)\right) = \int_{\mathbf{z}} q\left(\boldsymbol{z}^{(h)}\right)\log q\left(\boldsymbol{z}^{(h)}\right) - \int_{\mathbf{z}} q\left(\boldsymbol{z}^{(h)}\right)\log p\left(\boldsymbol{z}^{(h)}\right), \tag{4.57}$$

$$\mathrm{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau})) = \sum_{i=L+1}^{N}\left[(a_i-\alpha)\psi(a_i) + \log\frac{\Gamma(\alpha)}{\Gamma(a_i)} + \alpha\log\frac{b_i}{\beta} + a_i\frac{\beta-b_i}{b_i}\right]. \tag{4.58}$$

The REVARB-*t* model selection step is performed by jointly optimizing all the model's hyperparameters and variational parameters by maximizing the lower bound on the marginal log-likelihood expressed in Eq. (4.48). The second and fourth lines in Eq. (4.48) show the only terms that include the observations $\boldsymbol{y}$. In those terms, similar to the GP-RLARX model in Section 4.2, the value of each output y_i is always weighted by the ratio a_i/b_i in the diagonal of the matrix $\boldsymbol{R}$, which comes from the expectation with respect to the gamma distribution $q(\boldsymbol{\tau})$ in Eq. (4.53). Observations which contain outliers are related to much lower ratio values than regular observations, not greatly interfering in the bound. Furthermore, similar to GP-RLARX, we can detect outliers in the estimation data by checking the values of such ratios for each training sample. Note again that the full REVARB-*t* bound in Eq. (4.48) is not factorizable along the observations.

4.4.1 Making Predictions with the REVARB-t Framework

We are interested in using the RGP-*t* model to perform free simulation, i.e., given a sequence of new inputs, to iteratively use only past inputs and past predictions to infer the next step output. Furthermore, each prediction should consist of a fully characterized distribution and consider the uncertainty of past predictions. Thus, predictions in the REVARB-*t* can be performed with a set of modified predictive equations from the original REVARB framework, presented in Section 3.3.1.

Given a new regressor vector $\hat{\boldsymbol{x}}_*^{(h)}$, the mean $\boldsymbol{\mu}_*^{(h)}$ and variance $\boldsymbol{\lambda}_*^{(h)}$ of the predictions $f_*^{(h)} \sim \mathcal{N}\left(\boldsymbol{\mu}_*^{(h)},\boldsymbol{\lambda}_*^{(h)}\right)$ within each layer of the RGP-*t* model are calculated by approximate propagating uncertainty between each layer, using the expressions below (see

Section 3.3.1 for a related derivation):

$$\mu_*^{(h)} = \left(\boldsymbol{B}^{(h)}\right)^\top \left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top, \tag{4.59}$$

$$\begin{aligned}\lambda_*^{(h)} = &\left(\boldsymbol{B}^{(h)}\right)^\top \left(\boldsymbol{\Psi}_{2*}^{(h)} - \left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top \boldsymbol{\Psi}_{1*}^{(h)}\right) \boldsymbol{B}^{(h)} \\ &+ \Psi_{0*}^{(h)} - \mathrm{Tr}\left(\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1} - \left(\boldsymbol{P}^{(h)}\right)^{-1}\right) \boldsymbol{\Psi}_{2*}^{(h)}\right),\end{aligned} \tag{4.60}$$

where we have defined the following matrices:

$$\boldsymbol{P}^{(h)} = \boldsymbol{K}_z^{(h)} + \sigma_h^{-2} \boldsymbol{\Psi}_2^{(h)}, \qquad 1 \leq h \leq H, \tag{4.61}$$

$$\boldsymbol{P}^{(H+1)} = \boldsymbol{K}_z^{(H+1)} + \boldsymbol{\Psi}_2^{'(H+1)}, \tag{4.62}$$

$$\boldsymbol{B}^{(h)} = \sigma_h^{-2} \left(\boldsymbol{P}^{(h)}\right)^{-1} \left(\boldsymbol{\Psi}_1^{(h)}\right)^\top \boldsymbol{\mu}^{(h)}, \qquad 1 \leq h \leq H, \tag{4.63}$$

$$\boldsymbol{B}^{(H+1)} = \left(\boldsymbol{P}^{(H+1)}\right)^{-1} \left(\boldsymbol{\Psi}_1^{'(H+1)}\right)^\top \boldsymbol{y}. \tag{4.64}$$

In the former expression for $\boldsymbol{B}^{(H+1)}$ in Eq. (4.64), the training observations $\boldsymbol{y}$ are once again weighted by the diagonal of the matrix $\boldsymbol{R}$, inside $\boldsymbol{\Psi}_1^{'(H+1)}$, which reduces the influence of the outliers in the predictions.

The terms $\Psi_{0*}^{(h)}$, $\boldsymbol{\Psi}_{1*}^{(h)}$ and $\boldsymbol{\Psi}_{2*}^{(h)}$ in Eqs. (4.59) and (4.60) are computed as before, but instead of the distributions $q\left(x_i^{(h)}\right)$ we use the new Gaussian approximation $q\left(x_*^{(h)}\right) = \mathcal{N}\left(x_*^{(h)} \middle| \mu_*^{(h)}, \lambda_*^{(h)} + \sigma_h^2\right)$ and replace $\boldsymbol{K}_f^{(h)}$ and $\boldsymbol{K}_{fz}^{(h)}$ respectively with $K_*^{(h)} = k\left(\hat{\boldsymbol{x}}_*^{(h)}, \hat{\boldsymbol{x}}_*^{(h)}\right)$ and $\boldsymbol{k}_{*z}^{(h)} = \left[k\left(\hat{\boldsymbol{x}}_*^{(h)}, \boldsymbol{\zeta}_1^{(h)}\right) \cdots k\left(\hat{\boldsymbol{x}}_*^{(h)}, \boldsymbol{\zeta}_M^{(h)}\right)\right]$.

The predicted variance $\lambda_*^{(H+1)}$ in the observation layer can be added by the median of the values $b_i/a_i|_{i=L+1}^N$ related to the training outputs, following our approach for the GP-RLARX model. Thus, the predicted moments in the last layer are then given by $\mathbb{E}\{y_*\} = \mu_*^{(H+1)}$ and $\mathbb{V}\{y_*\} = \lambda_*^{(H+1)} + \mathrm{median}(b_i/a_i|_{i=L+1}^N)$.

Algorithm 4 summarizes the use of the RGP-t/REVARB-t framework in the robust system identification task, where we highlight that at the end of the estimation step it is possible to check which estimation samples probably contain outliers. Besides, each output during free simulation on test data is approximated by a fully defined Gaussian distribution.

4.5 Experiments

In this section we reproduce the several computational experiments firstly reported by us in Mattos *et al.* (2017) in order to evaluate the proposed GP-RLARX and

Algorithm 4: REVARB-t for outlier-robust dynamical modeling with the RGP-t model.

- Estimation step

Require: $\boldsymbol{u} \in \mathbb{R}^N$ (external input), $\boldsymbol{y} \in \mathbb{R}^N$ (output), H (number of hidden layers), M (number of inducing points), L (latent order lag), L_u (input order lag)

Initialize kernel hyperparameters and variational parameters;

repeat

Compute the evidence lower bound with Eq. (4.48);

Compute the analytical gradients of Eq. (4.48) with respect to the unknown parameters;

Update parameters with a gradient-based method (e.g. BFGS);

until convergence or maximum number or iterations

Output the optimized parameters;

Check the ratios $a_i/b_i|_{i=L+1}^{N}$, the smallest values are related to the estimation samples which probably contain outliers;

- Free simulation with test data

Require: Test external inputs $\boldsymbol{u}_* \in \mathbb{R}^{N_*}$ and the previously estimated RGP-t model

for $i = 1 : N_*$ **do**

for $h = 1 : H$ **do**

Compute the predictive mean $\mu_{*i}^{(h)}$ and variance $\lambda_{*i}^{(h)}$ with Eqs. (4.59) and (4.60);

Update the variational distribution of the new latent dynamical variable with $q\left(x_{*i}^{(h)}\right) = \mathcal{N}\left(x_{*i}^{(h)} \middle| \mu_{*i}^{(h)}, \lambda_{*i}^{(h)} + \sigma_h^2\right)$;

end for

Compute the predictive mean $\mu_{*i}^{(H+1)}$ and variance $\lambda_{*i}^{(H+1)}$ of the output layer with Eqs. (4.59) and (4.60);

Output $y_{*i}^{(H+1)} \sim \mathcal{N}\left(\mu_{*i}^{(H+1)}, \lambda_{*}^{(H+1)} + \text{median}(b_i/a_i|_{i=L+1}^{N})\right)$;

end for

RGP-t models and compare them with the standard GP-NARX and the robust GP-tVB. The experiments include six artificial benchmarks available in the system identification literature and two datasets related to process industry systems, available in the DaISy (Database for the Identification of Systems) repository (MOOR, 2016). In the latter two cases we also include experiments with the standard RGP model with Gaussian likelihood.

4.5.1 Artificial Benchmarks

We use the five artificial datasets previously described in Tab. 7 and a sixth dataset, derived from the artificial plant presented by Campos *et al.* (2000), summarized in Tab. 9. The outliers were generated by sampling from $\boldsymbol{\sigma}(\boldsymbol{y}) \times \mathcal{T}(0,2)$, i.e., a Student-$t$ distribution with zero mean and 2 degrees of freedom scaled by the standard deviation

Table 9 – Details of the sixth artificial dataset used in the robust computational experiments. The other five datasets used in Section 4.5.1 are detailed in Tab. 7. Note that $\mathrm{U}(A,B)$ is a random number uniformly distributed between A and B.

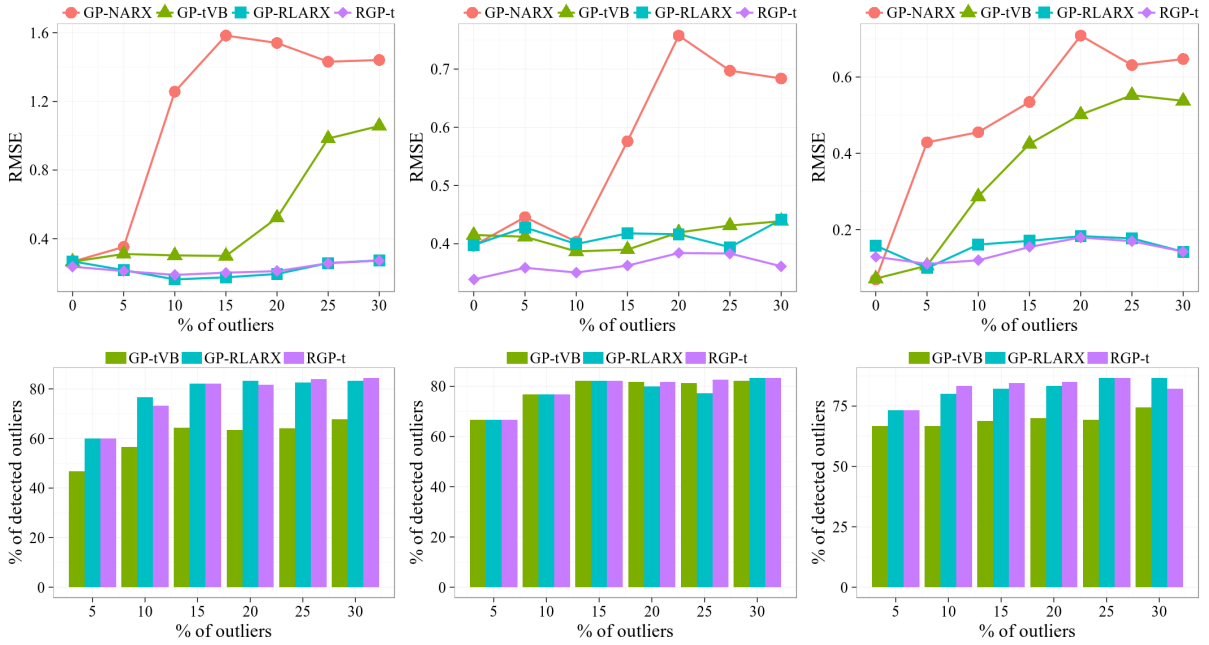
		Input/Samples		
#	**Output**	**Estimation**	**Test**	**Noise**
6	$x_i^{(1)} = x_{i-1}^{(2)}$ $x_i^{(2)} = -\frac{3}{16}\frac{x_{i-1}^{(1)}}{\left(1+\left(x_{i-1}^{(2)}\right)^2\right)} + x_{i-1}^{(2)} + u_{i-1}$ $y_i = x_i^{(1)}$	$u_i = \mathrm{U}(-0.5, 0.5)$ 300 samples	$u_i = \mathrm{U}(-0.5, 0.5)$ 300 samples	$\mathcal{N}(0, 0.0065)$

of the original estimation data[5]. We considered scenarios without outliers and with 5%, 10%, 15%, 20%, 25% and 30% of the estimation samples incrementally contaminated by outliers. In all experiments we used a RGP-t model with 1 hidden (transition) layer and 1 output (observation) layer. The number of inducing points M was fixed to 10% of the number of samples for both GP-RLARX and RGP-t models and the pseudo-inputs were initialized using the PAM (Partition Around Medoids) algorithm. We note that the results in this section should not be compared with the results previously reported in Section 4.1.3, since the datasets were regenerated.

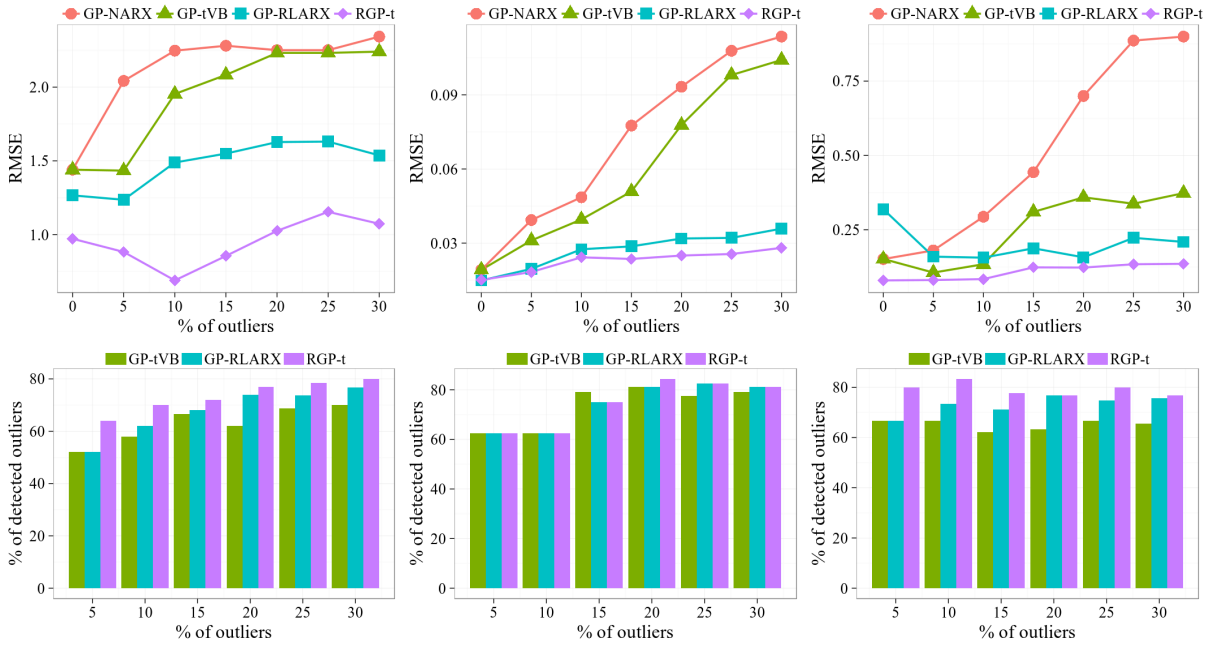
The obtained root mean square errors (RMSE) are presented in the line charts of Fig. 27. The robustness of the GP-RLARX model was already praised in Mattos *et al.* (2016) and its latent autoregressive structure was able to pair up with the heavy tailed Student-t likelihood and be more tolerant to outliers than GP-tVB. The RGP-t model maintains those features and incorporates additional representational capabilities to the dynamical modeling. Thus, besides the dataset *Artificial 1*, where RGP-t performed quite similar to GP-RLARX, in almost all the other scenarios the multilayered structure of RGP-t was able to obtain better results. For instance, in the case of *Artificial 2*, *4* and *6* datasets, great gain in performance occurred in all levels of contamination, which highlights the powerful hierarchical structure of the RGP-t model.

As mentioned before, the variational framework of the three robust models, i.e., GP-tVB, GP-RLARX and RGP-t, allows for a practical way of detecting estimation samples that contain outliers. After the model optimization step, by computing the ratios a_i/b_i related to the variational distribution $q(\boldsymbol{\tau})$ for the observation noise precisions along the estimation data and sorting the values, we can select the smallest ones, which

[5]Note that, since in the experiments of this section all the robust models apply a Student-t likelihood, we do not favor any one of them to the detriment of the others by sampling the outliers from a Student-t distribution.



(a) *Artificial 1* dataset. (b) *Artificial 2* dataset. (c) *Artificial 3* dataset.



(d) *Artificial 4* dataset. (e) *Artificial 5* dataset. (f) *Artificial 6* dataset.

Figure 27 – Line charts for the RMSE values related to the free simulation on test data with different levels of contamination by outliers. The correspondent bar plots indicate the percentage of outliers detected by the robust models using the variational framework.

correspond to smallest variational precisions (or largest variances), and associate them with samples containing outliers. The bar plots in Fig. 27 show the results of the application of such methodology, where we have selected in each case the expected amount of outliers (5% to 30%).

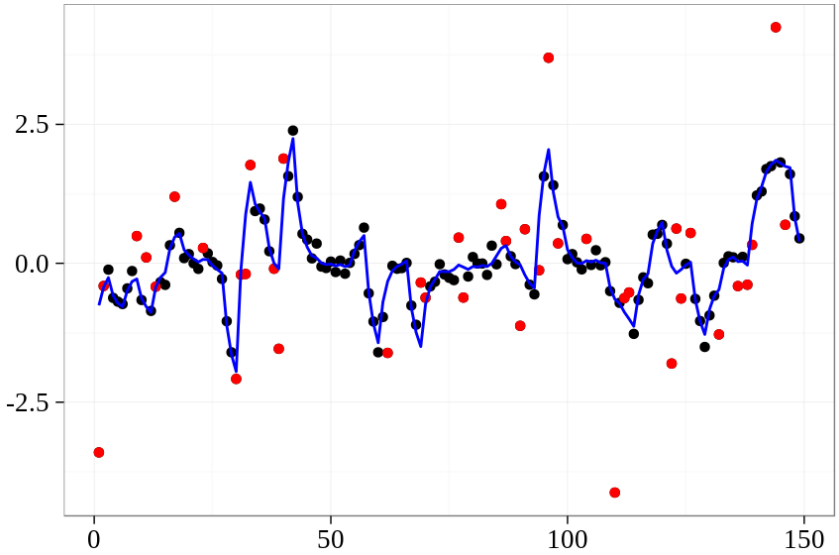


Figure 28 – Example of the robust filtering property of the GP-RLARX model. The **black** points are training observations with only Gaussian noise, while **red** points are outliers. The **blue** line indicates the mean values of the variational parameters after optimization, which act as filtered versions of the training data.

The three robust models are able to detect most of the outliers in the corrupted scenarios. Interestingly, although all the models use the same Student-t likelihood, in many cases the GP-RLARX and RGP-t were able to detect more outliers than GP-tVB, with RGP-t being slightly better overall, mainly for *Artificial 4* and *6* datasets. It should be noted that, in some scenarios, such as the ones of the *Artificial 5* dataset, although GP-tVB has been able to detect almost the same amount of outliers of the other robust models, its performance in terms of test prediction (see the correspondent line chart in Fig. 27) was considerably worse than GP-RLARX and especially RGP-t. Such observation once again stresses the importance of the latent autoregressive structure in the outlier-robust system identification task.

Before ending this subsection of experiments, we illustrate an interesting property of the GP-RLARX model. Since in its emission layer, expressed in Eq. (4.18), the latent output of the recurrent layer differs from the observed output only by the observation noise, an useful byproduct of the model optimization is a filtered version of the training data. An example is shown in Fig. 28 for the *Artificial 5* dataset with 25% of outliers. It is worth noticing that such property is not readily shared by the RGP-t model, since its outputs are obtained after passing the recurrent latent space through an additional nonlinear mapping learned in the separate observation layer.

Table 10 – RMSE and NLPD results for free simulation on test data after estimation on the *pH* dataset without and with outliers. The percentage of correctly detected outliers is also presented for the robust models that use variational-based inference.

	Without outliers		**With 30% of outliers**		
	RMSE	**NLPD**	**RMSE**	**NLPD**	**% detected**
GP-NARX	0.8716	1.8435	1.2735	2.3344	-
GP-tVB	0.9834	4264.0	1.0619	28.6278	76.7
GP-RLARX	0.8305	1.9181	0.7776	1.4777	75.0
RGP ($H=1$)	0.6658	1.7751	0.9299	1.8496	-
RGP ($H=2$)	**0.6616**	1.5625	1.2258	2.1177	-
RGP-t ($H=1$)	0.6661	**0.6363**	0.6221	1.4663	81.7
RGP-t ($H=2$)	0.6796	1.4283	**0.6192**	**1.1978**	**85.0**

4.5.2 pH Data

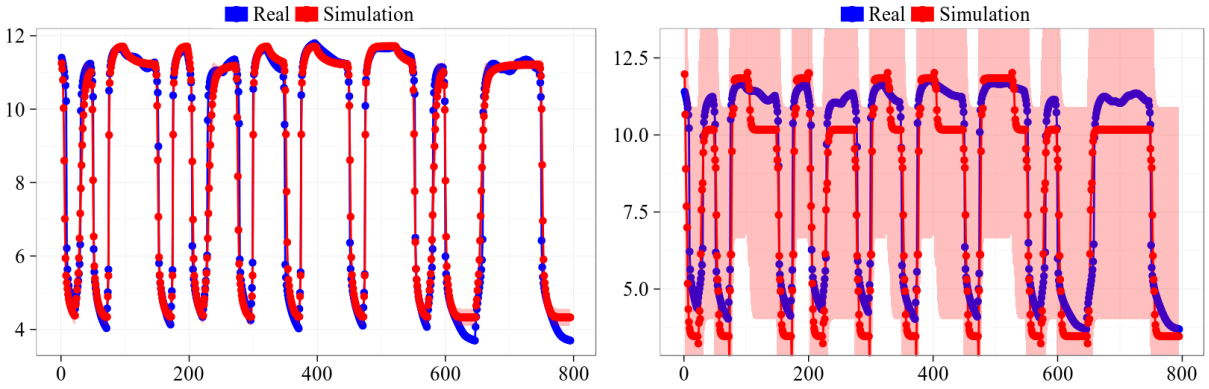
We now evaluate the GP models with the *pH* dataset[6]. The data comes from a pH neutralization process in a constant volume stirring tank. The control input is the base solution flow and the output is the pH value of the solution in the tank. We apply the first 200 samples for estimation and the next 800 samples for validation (testing). Two scenarios were considered, one without outliers and other where the estimation data was contaminated with 30% of outliers, sampled from $\boldsymbol{\sigma}(\boldsymbol{y}) \times \mathscr{T}(0,2)$, where $\boldsymbol{\sigma}(\boldsymbol{y})$ is the standard deviation of the original estimation data. We chose the orders $L = L_u = 5$ for the regressors. Besides the models used in Section 4.5.1, we also evaluate the original RGP model with Gaussian likelihood (with 1 and 2 hidden layers).

In Tab. 10 we report the test RMSE values and the average negative log-predictive density (NLPD). When applicable, the amount of outliers correctly detected by the robust models, using the variational strategy mentioned before, is also shown.

The corrupted scenario is quite extreme, with few estimation samples and a large contamination rate. Even though, the GP-RLARX model was better than the GP-tVB and both RGP-t models were able to reasonably learn the system dynamics. Actually, the RMSE values obtained for those models were even lower in the corrupted case than in the non-corrupted case, which can be explained by the fact that the former follows the expected Student-t noise prior of the robust models. Similar behavior was also observed in some of the results presented in Fig. 27 in Section 4.5.1.

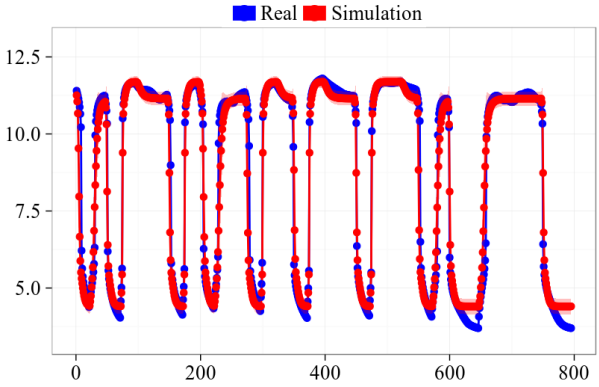
The free simulation outputs on test data for the best model, the RGP-t with 2

[6]Data available in the DaISy repository (MOOR, 2016) at <http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html>.

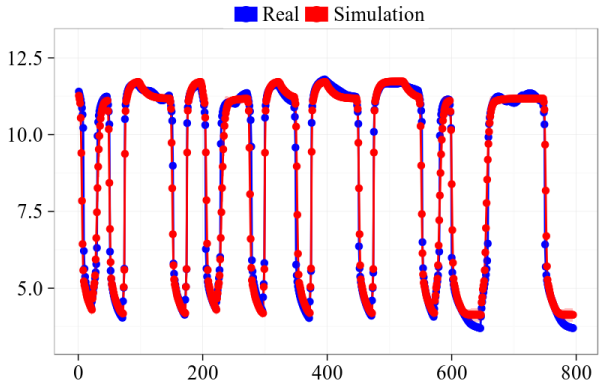


(a) RGP ($H = 2$) without outliers.

(b) RGP ($H = 2$) with 30% of outliers.

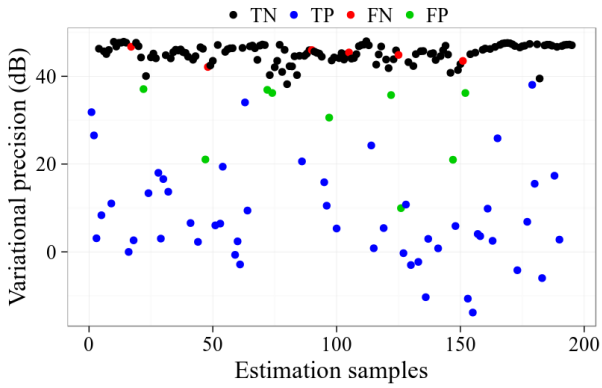


(c) RGP-t ($H = 2$) without outliers.

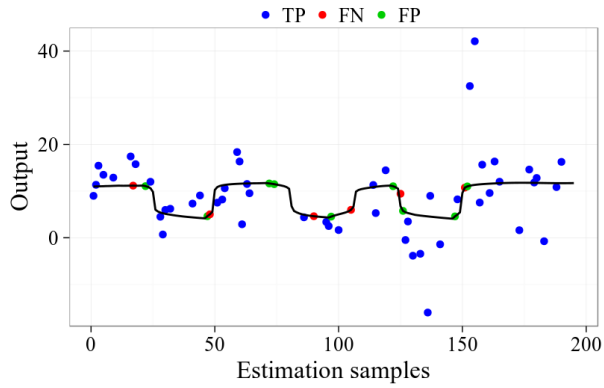


(d) RGP-t ($H = 2$) with 30% of outliers.

Figure 29 – Free simulation on test data after estimation on the pH train dataset without and with outliers. The shaded areas indicate ±2 standard deviations around the predicted mean values.



(a) Variational precisions after model optimization.



(b) Detected outliers.

Figure 30 – Outlier detection by the RGP-t model with 2 hidden layers and REVARB-t inference for the pH estimation data in the scenario with 30% of outliers. TN are the true negatives (correctly classified as non-outliers), TP are the true positives (correctly classified as outliers), FN are the false negatives (undetected outliers) and FP are the false positives (regular samples misclassified as outliers). In the right side, the black line indicates the original training output.

hidden layers, and its non-robust counterpart, RGP with 2 hidden layers, are presented in Fig. 29, where the shaded areas indicate ±2 standard deviations around the predicted

mean values. The RGP-t model seems "overconfident", since its shaded area is barely visible, but it still has presented superior performance for all three considered metrics in Tab. 29 for the corrupted scenario. Note that the output of the non-robust RGP model, although much worse in the case with outliers, is not so "wild" due to the presence of the external input, which is not contaminated.

In Fig. 30 we report results on the outlier detection capability of the RGP-t model with 2 hidden layers and REVARB-t inference. On the left side the variational precisions, i.e., the ratios a_i/b_i, are shown, where we can see a clear difference in the magnitude of the optimized precisions between outliers and regular samples. On the right side we confirm that all the severe outliers were correctly detected (true positives, represented by the blue dots), since the undetected ones (false negatives, red dots), are actually not far away from the original data. The points misclassified as outliers (false positives, green dots), whose amount was not enough to compromise the performance of the model, are also shown.

4.5.3 Heat Exchanger Data

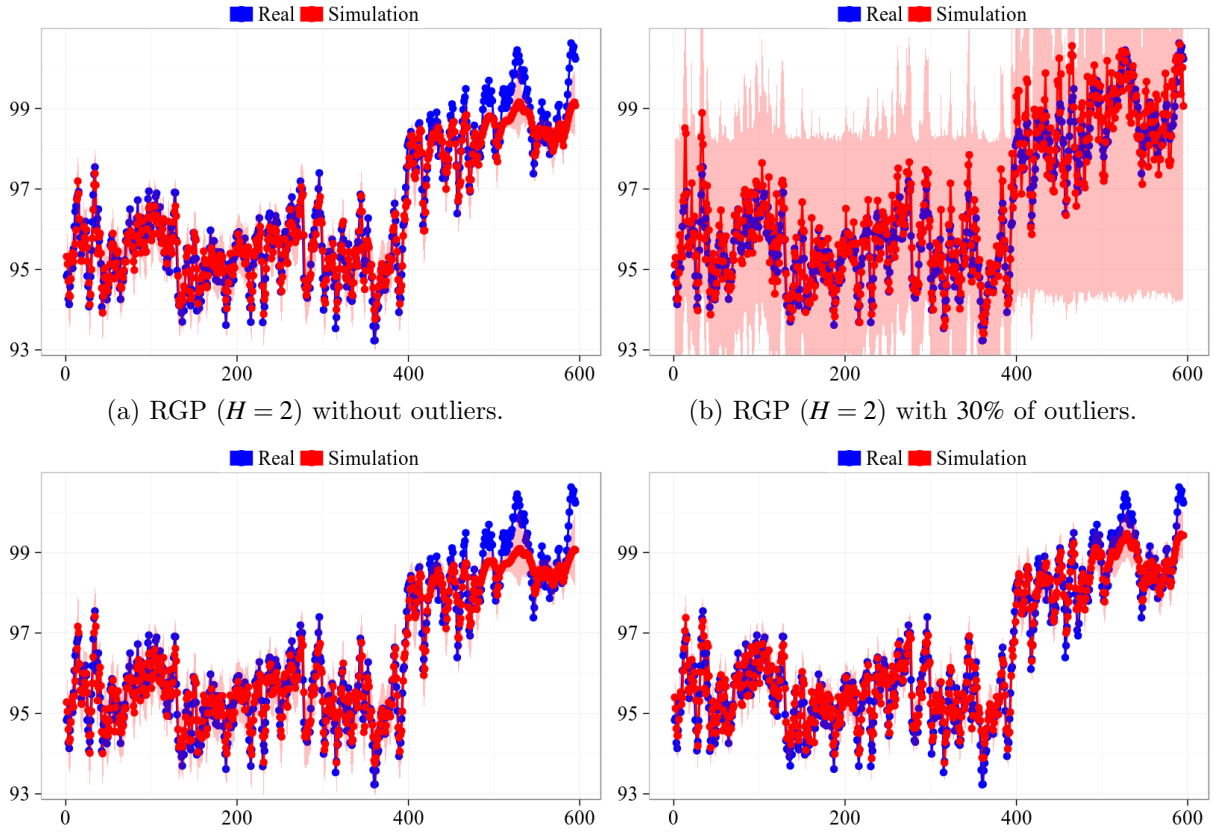
We conclude the experiments with the *Heat Exchanger* dataset[7]. The data comes from a liquid-saturated steam heat exchanger, where water is heated by pressurized saturated steam through a copper tube. The output variable is the outlet liquid temperature and the input variable is the liquid flow rate. One more time we applied the GP models to a scenario without outliers and another with 30% of outliers, which were sampled from the scaled distribution $\boldsymbol{\sigma}(\boldsymbol{y}) \times \mathscr{T}(0,2)$. The estimation set contains 300 samples (starting from the 101-th sample, since the first 100 are constant), while the test set contains the next 600 samples. We fixed the orders $L = L_u = 5$ for the regressors.

The obtained RMSE and NLPD values are shown in Tab. 11. The GP-RLARX presented good results, especially when compared to GP-tVB, but the robustness of both RGP-t models was again verified, with the 2-layered variant presenting the best RMSE and NLPD metrics. The outputs of the free simulation test for the latter model and its non-robust RGP counterpart are presented in Fig. 31, where we can observe that the RGP-t is unaffected by the outliers and actually present the best performance. One

[7]Data available in the DaISy repository (MOOR, 2016) at <http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html>.

Table 11 – RMSE and NLPD results for free simulation on test data after estimation on the *Heat Exchanger* dataset without and with outliers. The percentage of correctly detected outliers is also presented for the robust models that use variational-based inference.

	Without outliers		**With 30% of outliers**		
	RMSE	**NLPD**	**RMSE**	**NLPD**	**% detected**
GP-NARX	0.4816	0.8360	1.7269	2.0772	-
GP-tVB	0.7030	13.4332	0.5627	1.7514	**77.8**
GP-RLARX	0.5532	0.8979	0.4362	0.7374	76.7
RGP ($H=1$)	**0.4223**	**0.6893**	0.7885	1.5402	-
RGP ($H=2$)	0.4638	2.2295	0.5977	1.5324	-
RGP-t ($H=1$)	0.4745	1.6603	0.4233	0.8927	74.4
RGP-t ($H=2$)	0.4563	1.4855	**0.4087**	**0.7039**	74.4



(a) RGP ($H=2$) without outliers.

(b) RGP ($H=2$) with 30% of outliers.

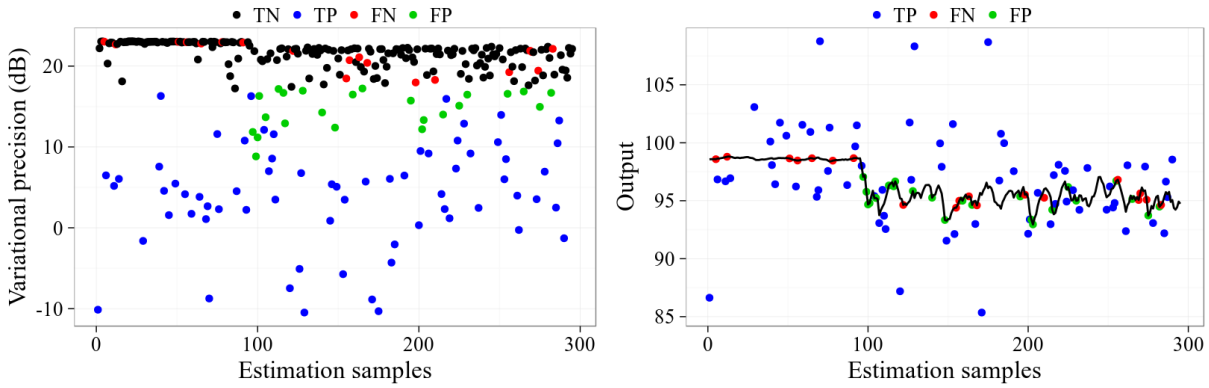
(c) RGP-t ($H=2$) without outliers.

(d) RGP-t ($H=2$) with 30% of outliers.

Figure 31 – Free simulation on test data after estimation on the *Heat Exchanger* train dataset without and with outliers. The shaded areas indicate ±2 standard deviations around the predicted mean values.

more time we note that the simulation of the non-robust RGP model in the contaminated scenario, although worse than before the outliers inclusion, is not completely far off the real output due to the presence of the uncontaminated exogenous input.

In Fig. 32 we show the outliers detected by the RGP-t/REVARB-t solution with 2 hidden layers. Even though this time it did not present the best detection rate



(a) Variational precisions after model optimization. (b) Detected outliers.

Figure 32 – Outlier detection by the RGP-*t* model with 2 hidden layers for the *Heat Exchanger* estimation data in the scenario with 30% of outliers. TN are the true negatives (correctly classified as non-outliers), TP are the true positives (correctly classified as outliers), FN are the false negatives (undetected outliers) and FP are the false positives (regular samples misclassified as outliers). In the right side, the black line indicates the original training output.

among the robust models, it is noticeable that most of the more relevant outliers were found. We emphasize that the low variational precision values (blue dots) shown in the left side are applied as weights to the correspondent output estimation samples, which greatly hinders any spurious effect caused by outliers on the model learning and predictive capabilities.

4.6 Discussion

In this chapter we have tackled the task of nonlinear system identification in the presence of outliers by introducing the GP-RLARX and RGP-*t* models, motivated by advances in robust GP-based modeling and approximate inference.

The GP-RLARX introduced the latent autoregressive structure which enables dynamical modeling without direct feedback of noisy observations possibly corrupted by outliers. The latter are handled by a heavy-tailed Student-*t* likelihood, which allows some probability far from the zero mean assumed for the observation noise and avoids model degradation during the estimation step. A variational approach is then derived to perform inference with such model.

By incorporating the robust Student-*t* treatment of the GP-RLARX model to the multilayer structure of the standard RGP, we get the RGP-*t* model, which applies a modified variational approach for inference, named REVARB-*t*. This more elaborated method enables us to perform robust learning within a powerful hierarchical recurrent

framework and opens up the possibility of applying such GP modeling approach to challenging nonlinear system identification scenarios.

We extensively evaluated the proposed models with computational experiments on several artificial benchmarks and datasets related to process industry systems. Some of the experiments were also performed with the GP-tVB (Student-t likelihood and variational inference) and GP-LEP (with a Laplace likelihood and EP inference) models, which present the original NARX dynamical structure and follow standard approaches to robust regression found in the literature. The better results presented by our proposed methods, which additionally incorporate dynamical latent structures, indicate that only the inclusion of a heavy-tailed likelihood is not enough to guard against outliers in an autoregressive set up, since they are still fed back as regressors in the inputs.

Although the GP-RLARX model was superior to both GP-tVB and GP-LEP models, the flexible and resilient structure of the RGP-t model was responsible for an impressive performance in almost all scenarios, even with large amounts of outliers. We also demonstrated how the variational approach paired with a Student-t likelihood can be applied to automatically detect outliers in the estimation samples and avoid their spurious effect on learning and prediction without directly removing them.

We conclude this chapter by noting that the superior performance obtained by the RGP-t model does not turn the GP-RLARX model obsolete. GP-RLARX's shallow structure and cleaner variational approach is computationally less demanding and allows for the filtering effect illustrated at the end of Section 4.5.1. The latter feature can even be more valuable than the actual predictions in some applications, for instance, if the task in hand demands a filtered version of the training data.

5 GP MODELS FOR STOCHASTIC DYNAMICAL MODELING

"Keep computations to the lowest level of the multiplication table."
(David Hilbert)

In Chapter 2 we commented how the standard GP modeling approach features a cubic $\mathcal{O}(N^3)$ complexity with respect to the number of training samples N, besides a $\mathcal{O}(N^2)$ memory requirement. In Section 2.6.1 we described a commonly used alternative, the variational sparse framework (TITSIAS, 2009a), which reduces such computational and memory demands to $\mathcal{O}(NM^2)$ and $\mathcal{O}(NM)$, respectively, where we can choose $M < N$ *inducing points*. However, the dependency with the number of samples remains and in larger N scenarios even the sparse approach may become infeasible.

The problem of modeling large quantities of sequential records generated by large scale systems and modern sensing devices has recently caught the attention of the dynamical modeling community (CARLI *et al.*, 2012; KIM *et al.*, 2013; CHENG *et al.*, 2015; GREEN *et al.*, 2015; GREEN; MASKELL, 2017; SANTOS; BARRETO, 2017). A seminal work that tackled the issue of handling large datasets for (static) regression problems with GP-based models was the one presented by Hensman *et al.* (2013), which incorporates the general ideas behind *stochastic variational inference* (SVI), proposed by Hoffman *et al.* (2013), to the variational sparse GP framework, enabling experiments with hundreds of thousands of samples.

The standard REVARB algorithm presented in Chapter 3, Section 3.3, works in *batch*, i.e., all the N input/output samples and the variational parameters are updated at the same time at each optimization step. Furthermore, since it follows strategies from Titsias' variational sparse framework, it presents similar computational requirements. In this chapter, inspired by Hensman *et al.* (2013) and other recent works, we formulate a stochastic version for the REVARB method, named S-REVARB, which enables the application of the RGP model, described in Section 3.2, to large datasets. We derive two algorithms to implement the S-REVARB inference, named *Local* and *Global* S-REVARB. Afterwards, we evaluate both approaches in system identification benchmarks with up to 95,000 training samples, much larger than the datasets previously presented in Chapter 3 for the original REVARB framework.

5.1 Stochastic Optimization

It is worth beginning by summarizing the general stochastic optimization task. The main idea behind stochastic optimization methods lies in successively approximating an initial candidate $\boldsymbol{\theta}_1$ to the solution $\boldsymbol{\theta}_{\text{opt}}$ of a problem over several iterations t, such as that, in probabilistic terms, $\lim_{t\to\infty} \boldsymbol{\theta}_t = \boldsymbol{\theta}_{\text{opt}}$ holds (ROBBINS; MONRO, 1951).

Arguably, the most common approach to guide an initial candidate solution involves taking small steps in the direction of noisy gradients as follows[1]:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \alpha_i \boldsymbol{g}_{t-1}, \tag{5.1}$$

where α_t is a small *learning step* (also known as *learning rate*) and the vector $\boldsymbol{g}_{t-1}$ indicates an unbiased estimation of the gradient of an objective function $L(\cdot)$ that needs to be maximized with respect to the candidate solution $\boldsymbol{\theta}_{t-1}$, i.e., $\boldsymbol{g}_{t-1} \approx \frac{\partial L}{\partial \boldsymbol{\theta}_{t-1}}$. We note that such approximation, although unbiased, may actually be considerably rough.

In the context of stochastic learning, the estimated gradient $\boldsymbol{g}_{t-1}$ related to the parameters update in each optimization step can be computed from a small set (a *mini-batch*) of training samples or even a single data point, which enables learning with very large datasets.

Although simple, it is known that optimization methods that follow Eq. (5.1) converge to a local optimum (or a global optimum, if the objective function $L(\cdot)$ is convex), given some mild assumptions, such as that $\sum \alpha_t = \infty$ and $\sum \alpha_t^2 < \infty$ (BOTTOU, 1998). We refer the readers to the work by Bottou (2004) for a comprehensive review on the importance of stochastic optimization in the machine learning literature.

5.2 Stochastic Variational Inference with GP Models

The general stochastic variational inference (SVI) framework proposed by Hoffman *et al.* (2013) applies stochastic learning techniques to optimize a variational objective function, e.g., a lower bound to the model marginal likelihood. SVI follows the simplified steps below:

1. Sample a certain number of data points from the training set;

[1] Although not the focus of the present thesis, stochastic gradient-free approaches are also largely present in the literature. We mention for reference metaheuristic algorithms (TALBI, 2009) and Bayesian optimization (SNOEK *et al.*, 2012). Interestingly, the latter is usually GP-based.

2. Optimize local variational parameters (if they exist);
3. Form intermediate global variational parameters;
4. Stochastically update global variational parameters.

Hoffman *et al.* (2013) applied this approach to learn latent Dirichlet allocation and hierarchical Dirichlet processes topic models from datasets with millions of data points. From the aforementioned listed steps, we can see that we have at least two requirements in order to apply SVI: (i) a set of global parameters; (ii) a factorized variational objective that enables separation of the observations and, if they exist, the correspondent local variational parameters.

Standard GP models do not present any of those requirements. In contrast, the variational sparse GP framework includes inducing points $\boldsymbol{z} \in \mathbb{R}^M$, which act as global variables. This feature is easily seen in the graphical models illustrated in Fig. 6, back in Chapter 2. However, the sparse equations presented in Section 2.6.1 turn the observations dependent, which can be noticed from the variational sparse lower bound in Eq. (2.25), reproduced here for clarity:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \log \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, \sigma_y^2 \boldsymbol{I} + \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{K}_{fz}^\top) - \frac{1}{2\sigma_y^2}\operatorname{Tr}(\boldsymbol{K}_f - \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{K}_{fz}^\top), \tag{5.2}$$

where the first term in the right side is responsible for preventing a factorization along the observations, since it contains the expressions $\log\left|\sigma_y^2\boldsymbol{I} + \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{K}_{fz}^\top\right|$ and $\left(\sigma_y^2\boldsymbol{I} + \boldsymbol{K}_{fz}\boldsymbol{K}_z^{-1}\boldsymbol{K}_{fz}^\top\right)^{-1}$, which are not factorisable with respect to the training samples. Those remarks were made by Hensman *et al.* (2013), who noticed that such behavior is result of the exact marginalization of the inducing variables $\boldsymbol{z}$.

In order to enable SVI in the variational sparse GP framework, Hensman et al. derived a non-collapsed lower bound, i.e., a bound explicitly parametrized by a variational distribution $q(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{m}, \boldsymbol{S})$, with moments $\boldsymbol{m} \in \mathbb{R}^M$ and $\boldsymbol{S} \in \mathbb{R}^{M\times M}$, as follows:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \sum_{i=1}^{N}\left\{\log \mathcal{N}(y_i|\boldsymbol{k}_i^\top\boldsymbol{K}_z^{-1}\boldsymbol{m}, \sigma_y^2) - \frac{1}{2\sigma_y^2}[\boldsymbol{K}_f]_{ii} - \frac{1}{2}\operatorname{Tr}(\boldsymbol{S}\boldsymbol{\Lambda}_i)\right\} - \mathrm{KL}(q(\boldsymbol{z})||p(\boldsymbol{z})), \tag{5.3}$$

where $\boldsymbol{k}_i$ is the i-th column of $\boldsymbol{K}_{fz}^\top$ and $\boldsymbol{\Lambda} = \frac{1}{\sigma_y^2}\boldsymbol{K}_z^{-1}\boldsymbol{k}_i\boldsymbol{k}_i^\top\boldsymbol{K}_z^{-1}$. Hensman et al. emphasize that the derivatives of this new bound with respect to the moments of $q(\boldsymbol{z})$, which are given by

$$\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{X})}{\partial \boldsymbol{m}} = \frac{1}{\sigma_y^2}\boldsymbol{K}_z^{-1}\boldsymbol{K}_{fz}^\top\boldsymbol{y} - \boldsymbol{\Lambda}\boldsymbol{m},$$

$$\frac{\partial \log p(\boldsymbol{y}|\boldsymbol{X})}{\partial \boldsymbol{S}} = \frac{1}{2}\boldsymbol{S}^{-1} - \frac{1}{2}\boldsymbol{\Lambda},$$

show that the optimal values of the moments are obtained when those derivatives are set to zero, when we recover the expressions previously presented in Eqs. (2.29) and (2.30) in the context of the variational sparse framework. Thus, the bound expressed in Eq. (5.3) is always looser (or equal, if the optimal moments are used) than the original variational sparse lower bound.

The new variational bound in Eq. (5.3), factorized along the training samples, enables *online* (one sample per iteration) or mini-batch updates. The SVI-GP methodology consists in sampling from the training data, computing the noisy derivatives[2] of Eq. (5.3) with respect to the moments $\boldsymbol{m}$ and $\boldsymbol{S}$, the pseudo-inputs $\boldsymbol{\zeta}_j|_{j=1}^{M}$ and the kernel hyperparameters, and then proceed by taking a small step in the direction of the computed gradient. We note that in Hensman *et al.* (2013) the pseudo-inputs are actually kept fixed and the update steps follow the *natural gradient* direction, i.e., the estimated gradient is weighted by the inverse of the Fisher information matrix (AMARI, 1998). Later, in Hensman *et al.* (2015), such SVI approach was adapted to highly scalable GP models for classification, where it was used to train models with millions of data points, a feat not possible with either standard or sparse GPs optimized in batch.

Recently, other authors have build upon the original SVI-GP framework to enable further scalability within the SVI context, such as the stochastic variational Kronecker-structured GP named *Blitzkriging* (NICKSON *et al.*, 2015), automated variational inference for non-Gaussian likelihood (DEZFOULI; BONILLA, 2015), a distributed SVI-GP framework (HOANG *et al.*, 2015) and hybridization with deep neural networks (WILSON *et al.*, 2016b).

One can see from the SVI-GP bound presented in Eq. (5.3) that it does not contain local latent variables such as the ones from the GP-LVM formulation (see Section 2.7). Hensman *et al.* (2013) comment that a similar factorized bound could be achieved for the GP-LVM and that a SVI approach would be possible by alternating between the optimization of the local variational parameters, i.e., the moments of the variational distribution $q(\boldsymbol{X})$ associated with the latent variables $\boldsymbol{X}$, while keeping the global parameters fixed, and then holding the moments of $q(\boldsymbol{X})$ and stochastically updating the global parameters. However, the authors did not presented any experimental evaluation of this methodology.

[2]We emphasize that the noise in the gradients comes from the fact that they are computed only from a subset of the training data in a given iteration of the algorithm.

Such experiments were performed in the thesis by Damianou (2015) in the context of high-dimensional data visualization, where SVI was successfully applied to the Bayesian GP-LVM formulation by Titsias and Lawrence (2010) in order to be used with datasets containing tens of thousands of samples, following an optimization methodology which includes an adaptive learning rate, firstly proposed by Hensman *et al.* (2014). Nevertheless, Damianou comments that, although viable, the experiments indicated that *the corresponding optimization procedure is very unstable because certain data batches can cause bifurcation-like effects.*

In Bui and Turner (2015) the authors argue that such SVI scheme for models derived from the GP-LVM is not practical for datasets of even modest size, since it can take a long time to converge because each iteration only updates the local variational parameters related to the samples in the current mini-batch and ignores the local optimizations performed in the previous iterations. In order to fix that, they propose the use of *recognition models*, for instance, a MLP neural network, to model the local variational parameters. In that case the network's parameters (weights) would be shared between all data points and actually behave themselves as global parameters. Bui and Turner illustrate the use of this approach, which does not contain any local parameter, in the task of unsupervised learning, using datasets with up to 18,000 images.

All those aforementioned contributions are not specifically designed to handle dynamical data and do not consider dynamical latent variables, such as the ones introduced by the RGP model, as described in Chapter 3. Thus, inspired by the referred recent works on SVI for GPs and GP-LVMs, we develop a non-collapsed variational lower bound for the REVARB algorithm, which enables stochastic inference with the RGP model in large data scenarios. The henceforward called S-REVARB framework will be presented in the next sections.

Remark Although not explored in the present thesis, it is worth mentioning additional recent works that aim to scale GP-based models to very large datasets, following alternative approaches, such as the use of mixtures of local experts (NGUYEN; BONILLA, 2014), tree-structured approximations (BUI; TURNER, 2014), distributed computations (GAL *et al.*, 2014; DAI *et al.*, 2014; DEISENROTH; NG, 2015; HOANG *et al.*, 2016), Kronecker and Toeplitz methods with kernel interpolation (WILSON; NICKISCH, 2015; WILSON *et al.*, 2015), fast matrix factorization techniques (AMBIKASARAN *et al.*, 2016), exploitation

of GP spectral representation (HENSMAN *et al.*, 2016) and parametric approximations (RAISSI, 2017).

5.3 S-REVARB: A Stochastic REVARB Framework

The S-REVARB framework modifies the original REVARB approach in order to enable stochastic variational inference with the RGP model. To this objective, we first need to factorize the REVARB lower bound across the training observations. We can do this by continuing from original REVARB's compact bound in Eq. (3.25), reproduced here for the sake of clarity:

$$\log p(\boldsymbol{y}) \geq \sum_{i=L+1}^{N} \sum_{h=1}^{H+1} \mathscr{L}_i^{(h)} + \sum_{i=1}^{N} \sum_{h=1}^{H} \mathscr{H}_i^{(h)} + \sum_{i=1}^{L} \sum_{h=1}^{H} \mathscr{L}_{0i}^{(h)} - \sum_{h=1}^{H+1} \mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right) \middle|\middle| p\left(\boldsymbol{z}^{(h)}\right)\right), \quad (5.4)$$

where each term in Eq. (5.4) is given by

$$\mathscr{L}_i^{(H+1)} = \left\langle p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H+1)}\right) \log p\left(y_i \middle| f_i^{(H+1)}\right) \right\rangle_{q(\boldsymbol{x})q(\boldsymbol{z})}, \quad (5.5)$$

$$\mathscr{L}_i^{(h)} = \left\langle p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right) \log p\left(x_i^{(h)} \middle| f_i^{(h)}\right) \right\rangle_{q(\boldsymbol{x})q(\boldsymbol{z})}, \quad (5.6)$$

$$\mathscr{H}_i^{(h)} = -\left\langle \log q\left(x_i^{(h)}\right) \right\rangle_{q(\boldsymbol{x})}, \quad (5.7)$$

$$\mathscr{L}_{0i}^{(h)} = \left\langle \log p\left(x_i^{(h)}\right) \right\rangle_{q(\boldsymbol{x})}, \quad (5.8)$$

$$\mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right) \middle|\middle| p\left(\boldsymbol{z}^{(h)}\right)\right) = \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right) \log q\left(\boldsymbol{z}^{(h)}\right) - \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right) \log p\left(\boldsymbol{z}^{(h)}\right). \quad (5.9)$$

Following Hensman *et al.* (2013), we proceed without optimally marginalizing the inducing points $\boldsymbol{z}^{(h)}$ in each layer h. Instead, we maintain the parametrized distributions $q\left(\boldsymbol{z}^{(h)}\right) = \mathscr{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{m}^{(h)}, \boldsymbol{S}^{(h)}\right), 1 \leq h \leq H+1$, in order to obtain a non-collapsed bound. Fortunately, we do not need to directly parametrize the covariance matrices $\boldsymbol{S}^{(h)} \in \mathbb{R}^{M \times M}$, but only its triangular Cholesky factor $\boldsymbol{L}_z^{(h)} \in \mathbb{R}^{M \times M}$, such as that $\boldsymbol{S}^{(h)} = \boldsymbol{L}_z^{(h)} \left(\boldsymbol{L}_z^{(h)}\right)^{\top}$ holds, which contains only $\frac{1}{2}M(M+1)$ variational parameters.

After the marginalization of all latent variables and some algebraic manipulation,

detailed in the Appendix A.5, the final S-REVARB lower bound is given by

$$
\begin{aligned}
\log p(\boldsymbol{y}) \geq \sum_{i=L+1}^{N} \Bigg\{ & -\frac{1}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\left(\Psi_0^{i(H+1)} + y_i^2 - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\right)\right) \\
& + \frac{1}{\sigma_{H+1}^2} y_i \left(\boldsymbol{\Psi}_1^{i(H+1)}\right)^{\top}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{m}^{(H+1)} \\
& - \frac{1}{2\sigma_{H+1}^2}\operatorname{Tr}\left(\left(\boldsymbol{S}^{(H+1)} + \boldsymbol{m}^{(H+1)}\left(\boldsymbol{m}^{(H+1)}\right)^{\top}\right)\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\right) \\
& + \sum_{h=1}^{H}\Bigg[-\frac{1}{2}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_h^2}\left(\Psi_0^{i(h)} + \left(\mu_i^{(h)}\right)^2 + \lambda_i - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{i(h)}\right)\right) \\
& + \frac{1}{\sigma_h^2}\mu_i^{(h)}\left(\boldsymbol{\Psi}_1^{i(h)}\right)^{\top}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)} \\
& - \frac{1}{2\sigma_h^2}\operatorname{Tr}\left(\left(\boldsymbol{S}^{(h)} + \boldsymbol{m}^{(h)}\left(\boldsymbol{m}^{(h)}\right)^{\top}\right)\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{i(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\right) \\
& - \int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right)\Bigg]\Bigg\} \\
& - \frac{1}{2}\sum_{h=1}^{H+1}\Bigg[\operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{S}^{(h)}\right) + \left(\boldsymbol{m}^{(h)}\right)^{\top}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)} - M \\
& + \log\left|\boldsymbol{K}_z^{(h)}\right| - \log\left|\boldsymbol{S}^{(h)}\right|\Bigg] \\
& + \sum_{i=1}^{L}\sum_{h=1}^{H}\left[\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right) - \int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right)\right].
\end{aligned}
\tag{5.10}
$$

where the factorized statistics $\Psi_0^{i(h)} \in \mathbb{R}$, $\boldsymbol{\Psi}_1^{i(h)} \in \mathbb{R}^{M\times 1}$ and $\boldsymbol{\Psi}_2^{i(h)} \in \mathbb{R}^{M\times M}$ are given by

$$
\begin{aligned}
\Psi_0^{i(h)} &= \left\langle \left[\boldsymbol{K}_f^{(h)}\right]_{ii} \right\rangle_{q(\cdot)^{(h)}} \\
\boldsymbol{\Psi}_1^{i(h)} &= \left\langle \left[\boldsymbol{K}_{fz}^{(h)}\right]_{i} \right\rangle_{q(\cdot)^{(h)}} \\
\boldsymbol{\Psi}_2^{i(h)} &= \left\langle \left[\boldsymbol{K}_{fz}^{(h)}\right]_{i}\left[\boldsymbol{K}_{fz}^{(h)}\right]_{i}^{\top} \right\rangle_{q(\cdot)^{(h)}}
\end{aligned}
\quad \Rightarrow q(\cdot)^{(h)} = \begin{cases} q\left(\boldsymbol{x}^{(1)}\right), & \text{if } h = 1, \\ q\left(\boldsymbol{x}^{(h)}\right) q\left(\boldsymbol{x}^{(h-1)}\right), & \text{if } 1 < h \leq H, \\ q\left(\boldsymbol{x}^{(H)}\right), & \text{if } h = H+1, \end{cases}
\tag{5.11}
$$

which follows the same notation used to define the REVARB statistics in Eq. (3.27). The new expressions in (5.11) are similarly calculated, also following the detailing in the Appendix A.1, but considering only a single sample or a mini-batch.

We emphasize that the S-REVARB bound in Eq. (5.10) is fully factorized, since it explicitly separates terms associated with the local variational parameters, related to each observation i, and terms associated only with the model's global parameters. For instance, in the original REVARB bound, after solving the expectations related to the

Table 12 – Comparison of computational and memory requirements of some GP-based dynamical models with respect to the number of training samples N, the number of pseudo-inputs M, the mini-batch size B and the number of hidden layers H.

	Computational	**Memory**
GP-NARX	$\mathcal{O}(N^3)$	$\mathcal{O}(N^2)$
Variational Sparse GP-NARX	$\mathcal{O}(NM^2)$	$\mathcal{O}(NM)$
RGP/REVARB	$\mathcal{O}((H+1)NM^2)$	$\mathcal{O}((H+1)NM)$
RGP/S-REVARB	$\mathcal{O}((H+1)BM^2)$	$\mathcal{O}((H+1)BM)$

h-th layer and marginalizing the inducing points $\boldsymbol{z}^{(h)}$, the terms $\log\left|\boldsymbol{K}_z^{(h)}+\frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right|$ and $\left(\boldsymbol{K}_z^{(h)}+\frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}$ appear. Since the statistic $\boldsymbol{\Psi}_2^{(h)}$ must be computed with the whole dataset, the bound cannot be factorized along the data samples. Such dependencies do not occur in S-REVARB's full non-collapsed bound, as expressed in Eq. (5.10).

Following the general SVI framework (HOFFMAN *et al.*, 2013), if we consider a mini-batch $\mathscr{B}$, i.e., a set of B sequential indexes sampled from the training data, we can rewrite Eq. (5.4):

$$\log p(\boldsymbol{y}) \geq \frac{N-L}{B}\sum_{i\in\mathscr{B}}\left[\sum_{h=1}^{H+1}\mathscr{L}_i^{(h)}+\sum_{h=1}^{H}\mathscr{H}_i^{(h)}\right]+\sum_{i=1}^{L}\sum_{h=1}^{H}\mathscr{L}_{0i}^{(h)}-\sum_{h=1}^{H+1}\mathrm{KL}\left(q\left(\boldsymbol{z}^{(h)}\right)\middle\|p\left(\boldsymbol{z}^{(h)}\right)\right), \tag{5.12}$$

whose analytical gradients can be used to perform mini-batch stochastic optimization. Note that the factor $\frac{N-L}{B}$ that scales the terms related to the local variables is necessary to avoid the dominance of the terms associated only with the global parameters, i.e., the KL divergence terms. The intuition presented by Hoffman *et al.* (2013) is that for $B=1$ the bound is actually formed by replicating a single observation for the whole training set. In the case of mini-batches, the proportional scaling is followed. Such scaling factor would converge to 1 if all the data was to be used at once, where we would have $B=N-L$. Moreover, the terms $\mathscr{L}_{0i}^{(h)}$ need only to be computed when the mini-batch contains samples which directly depend on the initial conditions, i.e., the indexes $1\leq i\leq L$.

Tab. 12 summarizes computational and memory requirements of the RGP/S-REVARB framework when compared to other dynamical GP-based models used in this thesis, where our stochastic approach removes the scale dependency with the number of training samples N.

5.3.1 Local S-REVARB: Recurrent SVI

One possible approach to use the S-REVARB non-collapsed bound to estimate the variational parameters of the RGP model follows the iterative strategy suggested by Hensman *et al.* (2013), similar to the experiments performed by Damianou (2015) in the context of the GP-LVM, summarized below:

1. Sample a mini-batch of training data points.
2. Hold all the model parameters with the exception of the local variational parameters of $\mathcal{N}\left(x_i^{(h)} \middle| \mu_i^{(h)}, \lambda_i^{(h)}\right)$ related to the sampled points.
3. Optimize the variational parameters of previous step until convergence, using analytical gradients of the lower bound.
4. Hold all variational means and variances.
5. Perform stochastic update steps for the global parameters, i.e., kernel hyperparameters, pseudo-inputs $\boldsymbol{\zeta}_j^{(h)}|_{j=1}^{M}$ and the moments of $q\left(\boldsymbol{z}^{(h)}\right)$ in all layers.

The steps are repeated until the parameters stop changing significantly or for a fixed amount of iterations. Since Step 3 can be computationally intense, it may actually be executed only in some cycles, or *epochs*, through the dataset, e.g., after a sequence of few epochs, perform local optimization in all iterations of the next epoch. Alternatively, instead of performing Step 3 until convergence, it is possible to consider only a stochastic update as follows:

3) (alternative) Take a small step in the direction of the lower bound's gradients with respect to the local variational parameters related to the samples in the mini-batch.

In this latter case, which is usually more convenient to implement, the alternative Step 3 should be performed every iteration of the algorithm.

Although the described procedure maintains all the local variables, which still scale with N, they are optimized in mini-batches of size $B \ll N$, greatly reducing the computational cost and memory requirement in each iteration, as presented in Tab. 12. This enables the RGP model to be trained with much more samples than with standard REVARB. When applying the S-REVARB in the latter approach, i.e., with the alternative Step 3, it will be named as the *Local S-REVARB* framework, to emphasize the explicit presence of the local latent variables.

Remark It is important to note that the Local S-REVARB framework does not correspond

to a new model. It considers the same RGP model presented in Section 3.2, including all its probabilistic distributions and variables. Thus, the Local S-REVARB is proposed as a stochastic and more scalable alternative inference method to the original REVARB.

5.3.2 Global S-REVARB: Sequential Recognition Models for S-REVARB

Since the local variational parameters related to each training sample are still present in the solution detailed so far, it may suffer from the issues argued by Bui and Turner (2015), where those variables would be stochastically updated at most once per epoch, i.e., each local optimization does not incorporate any knowledge of other local updates. That could turn the optimization procedure slow and overall inefficient. Bui and Turner fix this issue by using MLP neural networks as recognition models. However, those do not directly consider dynamical data or the RGP structure.

In the case of S-REVARB, we can use the RNN sequential recognition model previously presented in Section 3.3.2 to generate the local variational parameters. In Section 3.3.2 it was applied in a batch context and only for the variational means. Here we are interested in the stochastic scenario and in also applying a separate recognition model for the local variances. Thus, for a given hidden layer $1 \leq h \leq H$, both recognition models are written as follows:

$$q\left(x_i^{(h)}\right) = \mathcal{N}\left(x_i^{(h)} \middle| \mu_i^{(h)}, \lambda_i^{(h)}\right), \tag{5.13}$$

$$\mu_i^{(h)} = g_\mu^{(h)}\left(\hat{\boldsymbol{x}}_{i-1}^{(h)}\right) = \boldsymbol{\phi}_{\mu,2}\left(\boldsymbol{W}_{\mu,2}^{(h)\top}\boldsymbol{\phi}_{\mu,1}\left(\boldsymbol{W}_{\mu,1}^{(h)}\hat{\boldsymbol{x}}_{i-1}^{(h)}\right)\right), \tag{5.14}$$

$$\lambda_i^{(h)} = g_\lambda^{(h)}\left(\hat{\boldsymbol{x}}_{i-1}^{(h)}\right) = \boldsymbol{\phi}_{\lambda,2}\left(\boldsymbol{W}_{\lambda,2}^{(h)\top}\boldsymbol{\phi}_{\lambda,1}\left(\boldsymbol{W}_{\lambda,1}^{(h)}\hat{\boldsymbol{x}}_{i-1}^{(h)}\right)\right), \tag{5.15}$$

where matrices $\boldsymbol{W}_{\mu,l}^{(h)}$ and $\boldsymbol{W}_{\lambda,l}^{(h)}$, $l \in \{1,2\}$, are the networks' weights, and $\boldsymbol{\phi}_{\mu,l}(\cdot)$ and $\boldsymbol{\phi}_{\lambda,l}(\cdot)$ denote element-wise activation functions. In this work we use networks with depth equal 2, i.e., with 1 hidden layer, but deeper networks could also be applied. The only restriction is that $g_\lambda^{(h)}$ must generate only non-negative values, since they are related to the variational variances. It is important to note that the latent variables in the input $\hat{\boldsymbol{x}}_{i-1}^{(h)}$ of the recognition models above are in practice replaced by their associated variational means $\mu_i^{(h)}$, since standard NNs do not directly handle probabilistic variables as inputs.

Fig. 33 illustrates the structure and connections of the recognition models that generates the variational means and variances in the hidden layers of the RGP model in this new approach. It is worth noticing that the recognition model related to

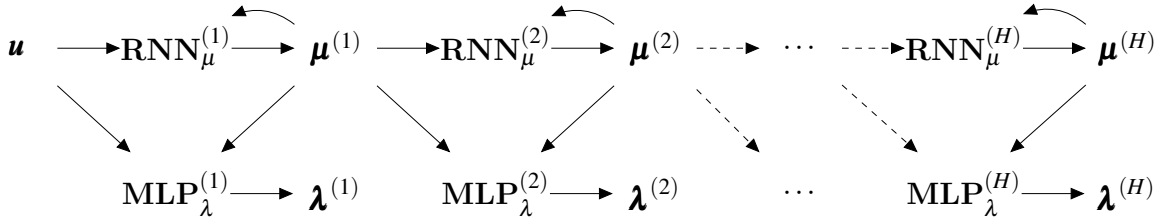


Figure 33 – Diagram for the MLP and RNN-based recognition models of the Global S-REVARB framework in a RGP model with H recurrent layers. Note that the outputs of the RNNs are used as inputs in the MLPs in the same and in the next hidden layer, which must be accounted when backpropagating gradients during training.

the variational means expressed in Eq. (5.14) is a RNN, while the model related to the variational variances in Eq. (5.15) is a regular feedforward MLP network. The MLP is trained with standard backpropagation (RUMELHART *et al.*, 1986), while the update of the RNN's parameters is made via the truncated backpropagation through time (TBPTT) algorithm (WILLIAMS; ZIPSER, 1995). Note that the outputs of the RNN are used as inputs in the MLP, so the gradients should be correctly propagated from one network to the other. Furthermore, when more than one hidden layer is present in the RGP model (e.g. 2 hidden recurrent layers), the gradients of a given layer should be backpropagated to the previous layer, following the dependencies shown in Fig. 33.

Importantly, the weights of the NNs are shared among the variational parameters, working as global parameters themselves. This is made clear when we recall that a stochastic update on the weights has effect in the output of the networks given any input. Such sequential recognition models when applied along S-REVARB's non-collapsed bound results in the *Global* S-REVARB framework, which enables mini-batch updates via stochastic gradient ascent. Since now the model contains only global variables, which do not scale with the number of samples, this approach opens up the possibility of performing inference with the RGP model in scenarios with even larger datasets.

Remark At a first glance, one could misinterpret the Global S-REVARB approach as a parametric version of the RGP model, due to the inclusion of the NN-based recognition models. However, this is not true. First, the obtained recurrent model is very different from a stand-alone parametric RNN, since it is built from the original RGP probabilistic modeling formulation by using the same analytical lower bound of the Local S-REVARB. Second, the sequential recognition models only act as *constraints* to the already analytical

nonparametric bound, functioning as variational (and not model) parameters. Similar arguments were also made by Dai *et al.* (2016) in the context of auto-encoded deep GP models.

5.3.3 Making Predictions with the S-REVARB Framework

In order to make predictions in the S-REVARB framework we can pursue the methodology presented by Girard *et al.* (2003) in the context of multi-step ahead GP predictions, which was also used in the original REVARB formulation.

Thus, we need to follow the same steps (and notation) detailed in Chapter 3, Section 3.3.1. First, we recall the conditional distribution of the sparse GP framework, reproduced below for the h-th layer:

$$
\begin{aligned}
p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right) &= \mathcal{N}\left(f_*^{(h)}\middle|\rho_*^{(h)},\varsigma_*^{(h)}\right), \\
\rho_*^{(h)} &= \boldsymbol{k}_{*z}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)}, \\
\varsigma_*^{(h)} &= K_*^{(h)} - \boldsymbol{k}_{*z}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{k}_{z*}^{(h)} + \boldsymbol{k}_{*z}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{S}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{k}_{z*}^{(h)}.
\end{aligned}
\tag{5.16}
$$

Note that the former conditional distribution contains the moments of the variational posterior $q\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)}\middle|\boldsymbol{m}^{(h)},\boldsymbol{S}^{(h)}\right)$, which in the case of S-REVARB are left explicitly parametrized by the moments $\boldsymbol{m}^{(h)}$ and $\boldsymbol{S}^{(h)}$, treated as variational parameters. We then proceed to compute the Gaussian approximation of the final predictive distribution in each layer:

$$
p\left(f_*^{(h)}\right) = \left\langle p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right)\right\rangle_{q(\boldsymbol{x}_*)} \approx \mathcal{N}\left(f_*^{(h)}\middle|\mu_*^{(h)},\lambda_*^{(h)}\right), \tag{5.17}
$$

$$
\begin{aligned}
\mu_*^{(h)} &= \left\langle \rho_*^{(h)}\right\rangle_{q(\boldsymbol{x}_*)}, \\
&= \left(\boldsymbol{B}^{(h)}\right)^\top\left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top,
\end{aligned}
\tag{5.18}
$$

$$
\begin{aligned}
\lambda_*^{(h)} &= \left\langle \varsigma_*^{(h)}\right\rangle_{q(\boldsymbol{x}_*)} + \mathbb{V}_{q(\boldsymbol{x}_*)}\left\{\rho_*^{(h)}\right\}, \\
&= \left(\boldsymbol{B}^{(h)}\right)^\top\left(\boldsymbol{\Psi}_{2*}^{(h)} - \left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top\boldsymbol{\Psi}_{1*}^{(h)}\right)\boldsymbol{B}^{(h)} + \Psi_{0*}^{(h)} \\
&\quad - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\left(\boldsymbol{I} - \boldsymbol{S}^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\right)\boldsymbol{\Psi}_{2*}^{(h)}\right),
\end{aligned}
\tag{5.19}
$$

where we have defined the matrix

$$
\boldsymbol{B}^{(h)} = \left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)}. \tag{5.20}
$$

The new statistics $\Psi_{0*}^{(h)}$, $\boldsymbol{\Psi}_{1*}^{(h)}$ and $\boldsymbol{\Psi}_{2*}^{(h)}$ are computed using the Gaussian approximation $q\left(x_*^{(h)}\right) = \mathcal{N}\left(x_*^{(h)} \middle| \mu_*^{(h)}, \lambda_*^{(h)} + \sigma_h^2\right)$. Finally, for the output layer we have $\mathbb{E}\{y_*\} = \mu_*^{(H+1)}$ and $\mathbb{V}\{y_*\} = \lambda_*^{(H+1)} + \sigma_{H+1}^2$.

It is worth emphasizing that the main difference between REVARB and S-REVARB in terms of the predictive step is that the latter, which considers a non-collapsed lower bound, applies the variational parameters $\boldsymbol{m}^{(h)}$ and $\boldsymbol{S}^{(h)}$ that explicitly parametrize the distribution $q\left(\boldsymbol{z}^{(h)}\right)$ to compute the predictive approximations. On the other hand, the original REVARB is able to apply the optimal moments of $q\left(\boldsymbol{z}^{(h)}\right)$, which are derived in its collapsed bound.

The aforementioned predictive equations can be readily applied by the Local S-REVARB, since it uses the local variational distribution $q\left(x_*^{(h)}\right)$ to compute the recursive predictions. In the case of the Global S-REVARB variant, we can describe at least two strategies for making predictions, listed as follows:

Variational sparse simulation The Local S-REVARB is able to approximately propagate the uncertainty during predictions by sequentially computing Eqs. (5.18) and (5.19) for each layer. The Global S-REVARB can also pursue the same methodology, named henceforth *variational sparse simulation*. Such strategy directly uses the moments of the distribution $q\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{m}^{(h)}, \boldsymbol{S}^{(h)}\right)$, from the variational sparse approximation, and the predictive variational distribution $q\left(\boldsymbol{x}_*^{(h)}\right)$. Note that in this approach the recognition networks are not used for making predictions.

Recognition-based simulation Since the Global S-REVARB includes NNs and RNNs, we have an alternative method to perform predictions with it that works as follows: the predictive means and variances in the hidden layers are directly computed using the recognition models, i.e., using Eqs. (5.14) and (5.15); then, in the output layer, the final prediction is made based on Eqs. (5.18) and (5.19), following the aforementioned variational sparse simulation approach. Such procedure takes advantage of the already learned sequential recognition models. However, it only considers the predicted variances of the last hidden layer ($h = H$) to perform the final prediction in the output layer. Thus, when using such *recognition-based simulation*, the uncertainty is computed locally but not propagated through the recurrent layers, only in the final observation layer.

After some preliminary experiments, we could not notice a clearly better

Algorithm 5: S-REVARB for stochastic dynamical modeling with the RGP model.

- Estimation step

Require: $\boldsymbol{u} \in \mathbb{R}^N$ (external input), $\boldsymbol{y} \in \mathbb{R}^N$ (output), H (number of hidden layers), M (number of inducing points), L (latent order lag), L_u (input order lag), B (mini-batch size)

Initialize kernel hyperparameters and variational parameters;

repeat

Sample B sequential points of the data;

Compute the evidence lower bound with Eq. (5.10) considering only the sampled points (scaling as in Eq. (5.12));

Compute the analytical gradients of Eq. (5.10) with respect to the unknown parameters;

Update parameters with a stochastic gradient method;

until convergence or maximum number or iterations

Output the optimized model;

- Free simulation with test data

Require: Test external inputs $\boldsymbol{u}_* \in \mathbb{R}^{N_*}$ and the previously estimated RGP model

for $i = 1 : N_*$ **do**

for $h = 1 : H$ **do**

Compute the predictive mean $\mu_{*i}^{(h)}$ and variance $\lambda_{*i}^{(h)}$ following one of the strategies presented in Section 5.3.3;

Update the variational distributions of the new latent dynamical variable with $q\left(x_{*i}^{(h)}\right) = \mathcal{N}\left(x_{*i}^{(h)} \middle| \mu_{*i}^{(h)}, \lambda_{*i}^{(h)} + \sigma_h^2\right)$;

end for

Compute the predictive mean $\mu_{*i}^{(H+1)}$ and variance $\lambda_{*i}^{(H+1)}$ of the output layer using Eqs. (5.18) and (5.19);

Output $y_{*i} \sim \mathcal{N}\left(\mu_{*i}^{(H+1)}, \lambda_{*i}^{(H+1)} + \sigma_{H+1}^2\right)$;

end for

methodology for performing predictions with the Global S-REVARB framework. Thus, we will explicit which approach was used for each experiment with the Global S-REVARB in the next sections.

5.3.4 Implementation Details

Algorithm 5 summarizes the general application of the S-REVARB framework to the RGP model. Both Local and Global versions, i.e., without or with sequential recognition models, follow the same general steps. Below we also list some of the implementation details we found useful when experimenting with S-REVARB. It is worth noting that the implementation details mentioned in Section 3.3.4 for the original REVARB still applies for its stochastic version.

Mini-batch sampling Since we are dealing with sequential training data, the mini-batch sampling procedure differs from the one followed by standard regression or classification methods. In our case, we slice the input and output time series with the size of the mini-batch (B) and shuffle the slices before presenting to the optimization algorithm. The important difference lies in the fact that we cannot shuffle the samples within the same mini-batch, in order to preserve the information about the system dynamics.

Activation functions We opted to use different activation functions for each layer in the recognition models of the Global S-REVARB. More specifically, the MLP that models the latent variances applies the hyperbolic tangent in the hidden units, which is given by $f(x) = \frac{\exp(2x)-1}{\exp(2x)+1}$, while the so-called SoftPlus activation function is used in the output units, which is given by $f(x) = \log(1+\exp(x))$. The latter function ensures that the recognition model will output only positive values for the variances. The RNN that models the variational means applies the ReLu (Rectified Linear Unit) activation function in the hidden units, which is simply given by $f(x) = \max(0,x)$ and is known for preventing vanishing or exploding gradient problems, which may rise in recurrent networks (GLOROT *et al.*, 2011). The output units apply the simple linear function $f(x) = x$, the usual choice in non-restricted regression tasks.

Model initialization In order to initialize the Global S-REVARB's recognition models, we follow the common approach of sampling the initial weights from the scaled Gaussian distribution $\sqrt{\frac{1}{N_{\text{in}}}}\mathcal{N}(0,1)$, where N_{in} is the number of inputs of the correspondent neuron. Since we use ReLu units in the hidden layer of the RNN models, we initialize their weights slightly different, sampling from $\sqrt{\frac{2}{N_{\text{in}}}}\mathcal{N}(0,1)$, which follows the recommendation made by He *et al.* (2015). Moreover, although we initialize all the neurons' biases with zeros, the biases of the ReLu units are initialized with small values, e.g., 0.01, in order to prevent "dead" units, i.e., which always output zero values, in the beginning of the optimization.

The components of the initial pseudo-inputs $\boldsymbol{\zeta}_j^{(h)}|_{j=1}^{M}$ are sampled from the normal distribution $\mathcal{N}(0,1)$, while the moments of the variational distributions $q\left(\boldsymbol{z}^{(h)}\right)$ are initialized with zero mean vectors and unitary diagonal covariance matrices, i.e., $\boldsymbol{m}^{(h)} = \boldsymbol{0}$ and $\boldsymbol{S}^{(h)} = \boldsymbol{I}$ at the start of the optimization, which implies the Cholesky factor $\boldsymbol{L}_z^{(h)} = \boldsymbol{I}$ for the matrix $\boldsymbol{S}^{(h)}$. The kernel hyperparameters are initialized as

follows: $\left(\sigma_f^{(h)}\right)^2 = 1$, $\left(w_d^{(h)}\right)^2 = 0.01$, $\sigma_h^2 = 0.01$. The jitter terms in each layer (see Section 3.3.4) are initialized with 0.01.

The hyperparameters and variational parameters of the Local S-REVARB are initialized following the original REVARB strategy, presented in Section 3.3.4, but subsampling the training data when necessary, e.g., for initializing the pseudo-inputs via clustering.

Optimization strategy There is a vast literature with algorithmic techniques and recommendations related to stochastic gradient-based optimization methods. In this thesis we use the ADAM optimizer, an adaptive strategy to the learning rate with separate values to each parameter being optimized (KINGMA; BA, 2015).

Moreover, we opted to decay the learning rate α_t in each t iteration of the optimization. More specifically, after 30% of the total iterations we start to exponentially decay α_t, from the initial value of α_{init} until the final value of α_{final}. We found empirically that the values $\alpha_{\text{init}} = 0.02$ and $\alpha_{\text{final}} = 0.002$ work well in practice, but those might need further tunning for other datasets.

We note that more elaborate adaptive learning rate strategies introduced for other non-GP SVI algorithms, such as the ones presented by Ranganath *et al.* (2013), Li and Ouyang (2016), were not explored in this work and is left to future investigations.

The TBPTT algorithm used to train the RNN within the Global S-REVARB follows gradients backpropagated across all samples of a given mini-batch and the previous 30 samples, counted backwards from the first sample in the mini-batch.

Finally, we hold the initial values of the kernel hyperparameters fixed for the initial iterations, e.g., the first 30% of the optimization steps, following similar recommendation by Hensman *et al.* (2013).

5.4 Experiments

In this section we will evaluate the RGP/S-REVARB solution in the task of system identification with large datasets. More specifically, we use the so-called *Silverbox* dataset and the *Wiener-Hammerstein* benchmark. We also perform an initial example with a smaller dataset in order to better analyze the proposed stochastic algorithms.

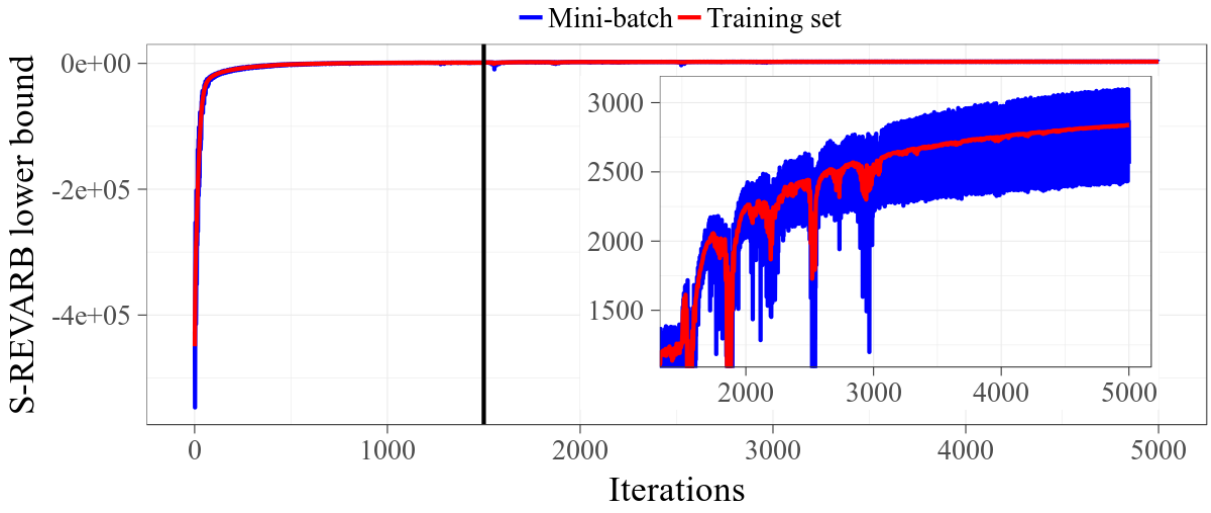
Table 13 – RMSE and NLPD values for the free simulation results of the S-REVARB framework on the *Damper* dataset. The Global S-REVARB followed recognition-based simulation. With the exception of the last two entries (separated by a horizontal line), all the values were previously reported in Tab. 3, Section 3.4.2.1, and reproduced here for convenience.

	RMSE	NLPD
Linear OE model (4th order)	27.1	-
Hammerstein-Wiener (4th order)	27.0	-
NARX (3rd order, wavelet)	24.5	-
NARX (3rd order, Tree partition)	19.3	-
NARX (3rd order, sigmoid network)	8.24	-
Standard GP-NARX	13.31	13.71
Variational Sparse GP-NARX ($M = 100$)	13.83	14.44
Reduced-rank GP-SSM (SVENSSON *et al.*, 2016)	8.17	3.71
SISOG (BIJL *et al.*, 2016)	7.12	NA
REVARB ($H = 1$)	11.18	3.47
REVARB ($H = 2$)	**6.04**	**3.05**
Local S-REVARB ($H = 2$)	8.30	3.51
Global S-REVARB ($H = 2$)	7.46	3.32

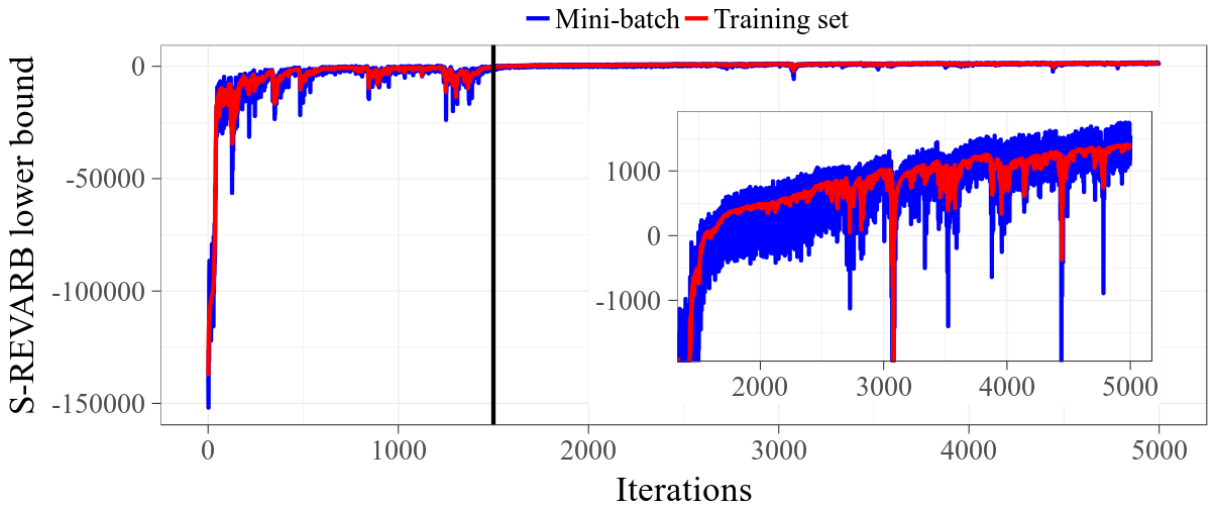
5.4.1 Initial Example

We first follow the experiments with the *Damper* dataset, previously used in Chapter 3, Section 3.4.2.1. We consider two RGP models containing $H = 2$ hidden layers and inference by the Local and Global S-REVARB algorithms. Both stochastic approaches use orders $L = L_u = 3$, $M = 100$ pseudo-inputs and mini-batch size $B = 200$, while the recognition models in the Global S-REVARB experiment have 100 hidden units and followed recognition-based simulation with the test data. The optimization via the Adam algorithm was executed for 5000 iterations.

Results are presented in Tab. 13, where we can see that both stochastic algorithms were able to obtain results not too far away from the batch REVARB. The Global S-REVARB was better than most other models presented in Tab. 3, Section 3.4.2.1, and reproduced in Tab. 13 for convenience, with the exception of the RMSE obtained by the SISOG method (BIJL *et al.*, 2016). The Local S-REVARB also presented a good result, with slightly worse RMSE than the reduced-rank GP-SSM (SVENSSON *et al.*, 2016), but better NLPD value. We note that the S-REVARB lower bound is looser than the collapsed REVARB bound, so it is somehow expected to obtain better results with the latter in scenarios with small and medium datasets, where it is feasible to use all training samples in the batch approach.



(a) Local S-REVARB.



(b) Global S-REVARB.

Figure 34 – Convergence curves of the S-REVARB lower bound with $H = 2$ hidden layers, in both Local and Global variants, during the training step on the *Damper* dataset using the Adam stochastic gradient algorithm. The vertical **black** line indicates the instant where the kernel hyperparameters are unfixed and the smaller pictures are zoomed versions of the curves after such instant.

Fig. 34 illustrates the convergence curves of the S-REVARB lower bound computed over the current mini-batch and over the entire training set using both stochastic inference strategies. The vertical black line indicates the instant where the kernel hyperparameters are unfixed and the smaller pictures are zoomed versions of the curves after such instant. We can see that the Local S-REVARB was able to obtain a higher value for the lower bound. This was the usual behavior in our experiments with smaller datasets, when several cycles (epochs) over the data are possible and the local unconstrained variational parameters seems to enable further optimization of the bound. However, we emphasize that

such additional optimization does not affect free simulation performance with test data, as can be noticed from the better predictive results obtained by the Global S-REVARB (Tab. 13), despite the lower value obtained for the bound.

Moreover, Fig. 34 also indicates that the Local S-REVARB presents a smoother convergence, while the Global S-REVARB is wigglier, which in the latter case is related to the update of all the weights of the NN recognition models in every iteration. Although in both cases the bound value in the mini-batch is noisier than the bound computed for the whole training data, they follow similar behavior, which indicates that the stochastic updates are valid approximations.

5.4.2 Stochastic System Identification with Large Datasets

We now perform experiments with two much larger system identification benchmarks. The *Silverbox* dataset was the subject of a special session organized at the IFAC Symposium on Nonlinear Control Systems (NOLCOS) in 2004 (SCHOUKENS *et al.*, 2003; WIGREN; SCHOUKENS, 2013)[3]. Its data comes from an electrical circuit describing the behavior of a mass-spring-damper nonlinear dynamical system with feedback, where the linear contributions are dominant. The input is related to the force applied to the mass and the output represents the mass displacement. A total of 91,072 samples are used for training and 40,000 for testing.

The *Wiener-Hammerstein* benchmark (SCHOUKENS *et al.*, 2009; HJALMARSSON *et al.*, 2012)[4], presented in the IFAC Symposium on System Identification in 2009 is comprised of data from an electronic nonlinear system consisting of a cascade of a linear dynamical block, a static nonlinear block and a final linear dynamical block. The training and test sets contain 95,000 and 84,000 samples, respectively. Note that we have skipped the first 5000 constant samples from the original 100,000 training observations and the last 4000 zero value test samples from the original 88,000.

Both the original *Silverbox* and *Wiener-Hammerstein* datasets are almost absent of observation noise, with the *Silverbox* having negligible noise and the *Wiener-Hammerstein* presenting a very high Signal-to-Noise Ratio (SNR) of 70dB. In order to

[3]Data available at <http://www.it.uu.se/research/publications/reports/2013-006/SNLA80mVZipped.zip>.

[4]Data available at <http://www.ee.kth.se/~hjalmars/ifac_tc11_benchmarks/2009_wienerhammerstein/WienerHammerBenchMark.mat>.

evaluate our approaches with noisier data, we add zero-mean Gaussian noise to both training sets with variances computed to turn their SNRs approximately equal to 20dB.

In all experiments we used the orders $L = L_u = 10$ and $M = 50$ pseudo-inputs. The mini-batch size was fixed to $B = 1000$, which showed empirically to be a good balance between stable gradients and computational cost. For the Global S-REVARB we use networks with 100 hidden units for the recognition models of both the variational means and variances.

For comparison, we have included in the experiments the variational sparse GP-NARX and RGP models with the standard REVARB method. In those cases we used only the first 5000 training samples for the estimation step. We also experimented with the same RNN applied as a sequential recognition model in the Global S-REVARB, used as a stand-alone model with both 1 and 2 hidden layers and 100 hidden units per layer. Those are initialized and stochastically optimized similarly to the Global S-REVARB algorithm.

The summary of obtained RMSE and NLPD values are presented in Tab. 14. In the case of the *Silverbox* dataset, the 1-hidden layer versions of all the multilayer models were better than their 2-hidden layer counterparts. This was somehow expected, since in this dataset the linear dynamics are dominant, as remarked by Marconato *et al.* (2012), and the inclusion of a second nonlinear recurrent layer proved to be more of a burden. The best result was obtained by the Global S-REVARB with $H = 1$, but the same model trained with the Local S-REVARB obtained the second best result.

Fig. 35 compares the absolute errors achieved by the Global S-REVARB solution with $H = 1$ and the 1-hidden layer RNN, where the former presents lower values in most areas. The final 5000 test samples are highlighted in the subfigure because they are related to inputs outside the range used to generate the training samples, which make them more difficult to predict. Still, the S-REVARB also presented smaller error values in that segment.

In the results for the *Wiener-Hammerstein* benchmark, also presented in Tab. 14, we had the opposite behavior. Most of the multilayer models with 2 hidden layers were better than their shallow versions. The exception was the Local S-REVARB. This time the best solution was the Global S-REVARB with $H = 2$. Fig. 36 presents the absolute test errors obtained by that best solution and the RNN with two hidden layers, where we can see that the largest errors are associated with the RNN model.

Table 14 – Summary of results for the free simulation on test data after estimation from large dynamical datasets. Note that only the stochastic methods (RNNs and S-REVARB variants) used the entire training sets, since they can be trained via mini-batches. The other models were optimized in batch using the first $N = 5000$ training samples.

Silverbox	RMSE	NLPD
RNN (1 hidden layer)	2.107×10^{-3}	-
RNN (2 hidden layers)	3.369×10^{-3}	-
Variational Sparse GP-NARX ($N = 5000$)	1.676×10^{-3}	-4.204
REVARB ($H = 1, N = 5000$)	1.757×10^{-3}	-4.217
REVARB ($H = 2, N = 5000$)	3.562×10^{-3}	-3.997
Local S-REVARB ($H = 1$)	1.245×10^{-3}	-4.222
Local S-REVARB ($H = 2$)	8.285×10^{-3}	-3.321
Global S-REVARB ($H = 1$)[a]	$\mathbf{1.052 \times 10^{-3}}$	$\mathbf{-4.226}$
Global S-REVARB ($H = 2$)[a]	1.4031×10^{-3}	-4.1436
Wiener-Hammerstein	RMSE	NLPD
RNN (1 hidden layer)	1.222×10^{-2}	-
RNN (2 hidden layers)	8.247×10^{-3}	-
Variational Sparse GP-NARX ($N = 5000$)	3.584×10^{-2}	-1.883
REVARB ($H = 1, N = 5000$)	2.037×10^{-2}	-2.406
REVARB ($H = 2, N = 5000$)	1.547×10^{-2}	-2.544
Local S-REVARB ($H = 1$)	1.295×10^{-2}	-2.609
Local S-REVARB ($H = 2$)	2.372×10^{-2}	-2.308
Global S-REVARB ($H = 1$)[b]	8.369×10^{-3}	-2.606
Global S-REVARB ($H = 2$)[b]	$\mathbf{5.664 \times 10^{-3}}$	$\mathbf{-2.643}$

[a] variational sparse simulation
[b] recognition-based simulation

Table 15 – Comparison of the number of adjustable parameters (RNNs) or hyperparameters and variational parameters (S-REVARB variants) in the experiments with the *Wiener-Hammerstein* benchmark ($N = 95{,}000$).

	Size
RNN (1 hidden layer)	2201
RNN (2 hidden layers)	4402
Local S-REVARB ($H = 1$)	194,206
Local S-REVARB ($H = 2$)	386,574
Global S-REVARB ($H = 1$)	8608
Global S-REVARB ($H = 2$)	15,378

The not so good results obtained by the Local S-REVARB in the experiments with the *Wiener-Hammerstein* dataset may be related to the the information provided by Tab. 15, which reports the number of adjustable parameters, hyperparameters and variational parameters present in the models that had access to the entire training set. As

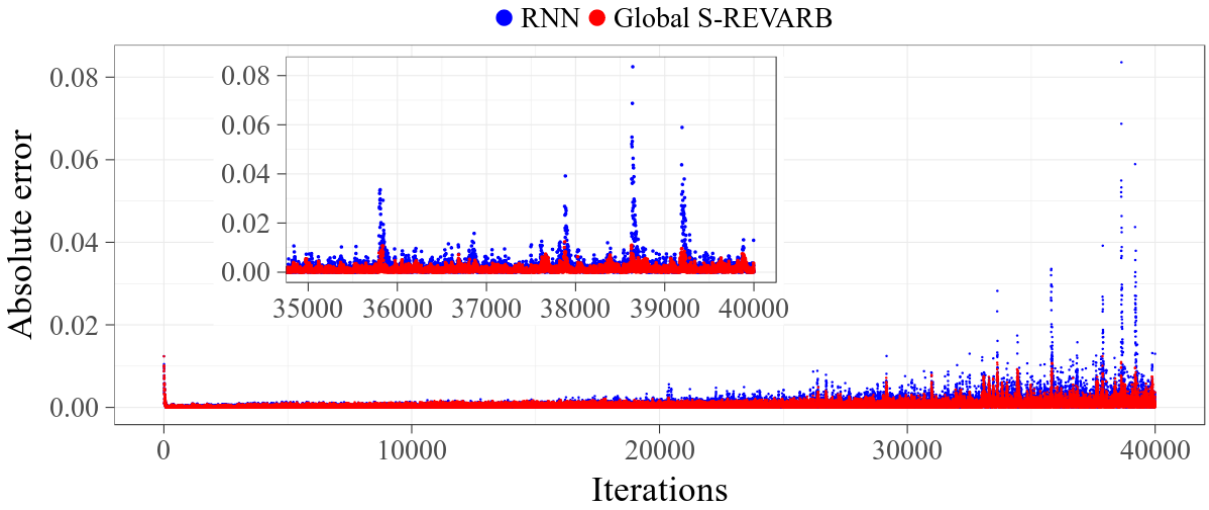


Figure 35 – Comparison between the absolute test errors obtained by the Global S-REVARB and the RNN on the *Silverbox* dataset, both with one hidden layer. The smaller picture is a zoomed version of the last 5000 test samples, the most difficult ones to predict correctly, since they were generated with inputs outside the range used during the training step.

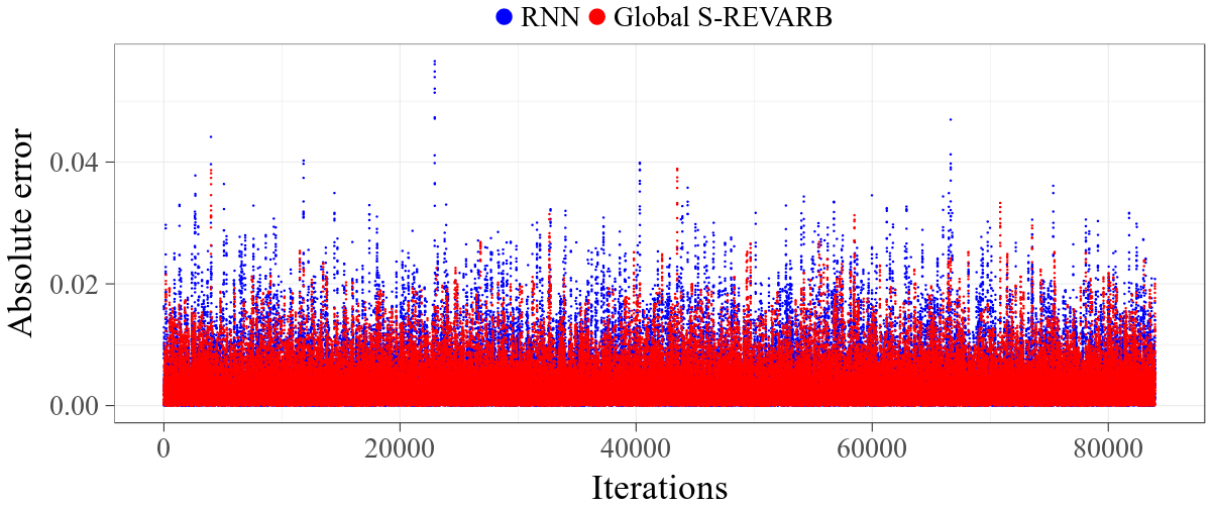


Figure 36 – Comparison between the absolute test errors obtained by the Global S-REVARB and the RNN on the *Wiener-Hammerstein* dataset, both with two hidden layers.

can be seen, the size of the Local S-REVARB implementation becomes considerably large with the increase of the number of training samples N, which may turn the optimization procedure more difficult, even though it is scalable in terms of computation and memory demands due to the mini-batch training. Moreover, since we have used 10,000 optimization iterations and mini-batches of $B = 1000$ size sampled from $N = 95{,}000$, each local variational parameter was stochastically update only 106 times. The good result obtained by the Local S-REVARB with one hidden for the *Silverbox* is possibly related to the simpler dynamics of that dataset, which did not require so many update steps from the initialization to get

a reasonable model.

On the other hand, the size of the Global S-REVARB implementation scales (linearly) only with the number of hidden layers and do not increase with the amount of available data, since there is no local variational parameters and the quantity of weights of the sequential recognition models does not grow with N. The great difference in size between the two S-REVARB methods, where the Local S-REVARB approach presents up to 25 times more adjustable parameters than its Global equivalent in Tab. 15, may justify the better performance of the latter in the reported results.

5.5 Discussion

In this chapter we have considered the challenge of learning dynamical models from large sequential datasets. The scalability issues of GP-based models, which even in their sparse versions still scale at least linearly with number of training samples, were tackled by following stochastic optimization procedures.

More precisely, inspired by recent advances on stochastic variational inference (SVI), we developed a non-collapsed factorizable modification to the original REVARB lower bound to the model marginal log-likelihood presented in Chapter 3, resulting in the S-REVARB framework, which aims for better scalable inference with the RGP model.

From such formulation, we described two stochastic learning algorithms, the Local S-REVARB and the Global S-REVARB. The former more directly applies the standard SVI ideas to the new factorized bound, keeping all the local variational parameters. The latter incorporates sequential recognition models in order to include NN-based constraints to the S-REVARB inference framework, which no longer contains local parameters and is even further scalable. Both algorithms enable optimization using noisy mini-batch updates via off-the-shelf stochastic gradient ascent strategies, such as the ADAM optimizer.

We evaluated the proposed solutions in the task of system identification from datasets with up to 95,000 training points, much more than which is computationally feasible with the original REVARB approach. The obtained results indicate that the stochastic variational framework is a viable scalable alternative to the REVARB batch approach, especially the Global S-REVARB variant, which avoids the increase of its implementation size with the number of training samples and has presented the best overall results.

In summary, this chapter greatly extends the possible scenarios where GP-based dynamical modeling can be applied, aiming for scalability without sacrificing the expressiveness of the hierarchical recurrent structure of RGP models.

6 CONCLUSIONS

> "The charges have to do with conspiracy to augment an artificial intelligence."
>
> (William Gibson, *Neuromancer*)

In recent years the machine learning and dynamical modeling communities have continued to share models, analyses and algorithms, presenting many interceptions and application subjects in common. In that sense, the present thesis considered the relevant problem of modeling dynamical systems only from their inputs and outputs, without the explicit knowledge of the internal physical functioning of the related phenomena, which characterizes the system identification task.

We pursued a Bayesian probabilistic approach to nonlinear system identification, using Gaussian Processes models to handle uncertainty, represent sequential data and make predictions via free simulation. In such context, we reviewed recent GP-based dynamical approaches and proposed the novel class of Recurrent Gaussian Processes models, specifically designed to treat dynamical data.

We tackled the challenging scenarios of learning dynamics directly from noisy data, data containing non-Gaussian noise in the form of outliers and learning from large datasets. In each case we have introduced novel models and inference methods, which were comprehensively evaluated and compared with other solutions available in the literature in several system identification benchmarks and other problems involving sequential records.

More specifically, we introduced the RGP/REVARB framework for general probabilistic recurrent modeling, the GP-RLARX and RGP-t/REVARB-t approaches to outlier-robust dynamical modeling and the S-REVARB approach, presented as two variants, the Local and Global S-REVARB algorithms, for scalable stochastic recurrent learning. In order to overcome the mathematical intractabilities of our models, we mainly followed variational procedures, i.e., deterministic (not based on sampling) methodologies which convert the inference step into an optimization problem. Tab. 16 summarizes the features presented by the dynamical GP models used in this work, where the entries marked with a "*" are our contributions.

The results obtained by the proposed probabilistic approaches throughout this thesis and the advantages they present over other kernel-based methods or parametric models are factors that justify the recent attention given to general GP modeling by the

Table 16 – Summary of features presented by some of the different dynamical GP-based models used in this work for nonlinear system identification. The models marked with a "*" are our contributions. All the listed approaches share the Bayesian nonparametric feature.

	Latent states	**Hierarchical**	**Propagates uncertainty**	**Likelihood**	**Inference**
GP-NARX				Gaussian	analytical
Sparse GP-NARX				Gaussian	variational
GP-LEP				Laplace	EP
GP-tVB				Student-*t*	variational
*GP-RLARX	✓		✓	Student-*t*	variational
*RGP	✓	✓	✓	Gaussian	REVARB/S-REVARB
*RGP-*t*	✓	✓	✓	Student-*t*	REVARB-*t*

research community and encourage us to continue pursuing GP-based models as a valuable approach to dynamical modeling in diverse practical learning scenarios.

6.1 Future Work

The versatility of the GP modeling framework and the expressiveness of the probabilistic recurrent models considered in this thesis are themes for further theoretical investigations and practical applications. Thus, we list below some interesting directions for additional research.

Alternative dynamical structures One could think of very different ways to represent the dynamical latent states of RGPs, such as the incorporation of derivative information (SOLAK *et al.*, 2003; AŽMAN; KOCIJAN, 2011), dynamics learned from the time steps (DAMIANOU *et al.*, 2011) or even alternative approaches based on the GP-LVM framework, with different priors and constraints (LAWRENCE; QUIÑONERO-CANDELA, 2006).

In terms of deep GP modeling, Duvenaud *et al.* (2014) present some recommendations for fixing pathologies that rise in deep architectures, such as the inclusion of direct links between the input and each layer of the model. We have made few preliminary experiments adapting that idea to RGPs, but the results were inconclusive and that approach should be further explored.

Moreover, if one aims to model long term temporal dependences, which is usually not required by the system identification tasks we have covered, the most promising works on RNNs indicate the need of some kind of *gating* mechanism to control the influence of past observations. This is the strategy pursued for instance by the powerful Long

Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) and Gated Recurrent Unit (GRU) (CHO *et al.*, 2014) networks, which have been widely applied to complex recurrent applications (SCHMIDHUBER, 2015). Such gating mechanism, which can be implemented in several different forms (GREFF *et al.*, 2016), could be adapted to the Bayesian approach of RGPs, which itself constitutes an interesting challenge.

Alternative inference implementation Our inference algorithms were implemented following analytical gradients calculated "by hand". This usually results in slightly faster codes, but it is tedious to derive and time consuming. Alternatively, instead of manually taking the gradients of the many variational lower bounds we have derived, one could use *automatic differentiation* tools to ease the implementation, such as the ones available in the Theano (Theano Development Team, 2016) or TensorFlow (ABADI *et al.*, 2016) programming frameworks. This purely practical issue is actually a very important step towards more widespread use of GP-based dynamical models.

Outlier-robust stochastic learning A notable absence in the Chapter 5 of this thesis, which covers stochastic learning for dynamical GPs, is the lack of a stochastic scalable version of the robust REVARB-t framework. Although we have tried to tackle this additional modeling scenario, we could not develop a coherent approach to deal with the issue of the variational parameters related to the noise precisions, as presented in the REVARB-t formulation in Section 4.4, Chapter 4. A prototype based on the Local S-REVARB algorithm did not work well, since it would take too many iterations until an outlier-corrupted observation could be detected and suppressed by the stochastic algorithm. A robust version of the Global S-REVARB also turned out to be difficult to conceive, since such variational parameters could not be modeled by a third recognition model, for they may be related to outliers, which do not present any "learnable" pattern by definition. A frequentist alternative could be pursued, such as the use of M-estimation methods (HUBER, 2011), but a Bayesian solution would be preferable.

Predictive control As a complement to the nonlinear system identification task, focus of this thesis, a compelling application for the presented dynamical GP models is within Model Predictive Control (MPC) methodologies. Although GP-based MPC has been

studied to some extent (KOCIJAN *et al.*, 2004; LIKAR; KOCIJAN, 2007; ROCHA *et al.*, 2016; KLENSKE *et al.*, 2016), it would be interesting to apply the proposed models, especially the outlier-robust GP-RLARX and RGP-t, to challenging control scenarios, where uncertainty propagation becomes even more relevant. Moreover, as described in the Appendix A.1, the variational frameworks we have presented can incorporate uncertain exogenous control inputs with minimal modifications, which enables their use with probabilistic controllers and widens even more their applicability.

BIBLIOGRAPHY

ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M. *et al.* TensorFlow: A system for large-scale machine learning. In: **Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)**. Savannah, Georgia, USA: USENIX, 2016.

ACKERMANN, E. R.; VILLIERS, J. P. D.; CILLIERS, P. Nonlinear dynamic systems modeling using Gaussian processes: Predicting ionospheric total electron content over south africa. **Journal of Geophysical Research: Space Physics**, Wiley Online Library, v. 116, n. A10, 2011.

AGGARWAL, C. C. **Outlier Analysis**. USA: Springer Science & Business Media, 2013.

AGUIRRE, L. A. **Introdução à identificação de sistemas – Técnicas lineares e não-lineares aplicadas a sistemas reais**. 3rd. ed. Belo Horizonte, MG, Brazil: Editora UFMG, 2007.

AL-SHEDIVAT, M.; WILSON, A. G.; SAATCHI, Y.; HU, Z.; XING, E. P. Learning scalable deep kernels with recurrent structure. **arXiv preprint arXiv:1610.08936**, 2016. Available on: <https://arxiv.org/abs/1610.08936>.

ALVAREZ, M.; PETERS, J. R.; LAWRENCE, N. D.; SCHÖLKOPF, B. Switched latent force models for movement segmentation. In: **Advances in Neural Information Processing Systems 23 (NIPS)**. Vancouver, Canada: NIPS Foundation, 2010. p. 55–63.

ALVAREZ, M. A.; LUENGO, D.; LAWRENCE, N. D. Latent force models. In: **Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Clearwater Beach, FL, USA: JMLR.org, 2009. v. 12, p. 9–16.

AMARI, S.-I. Natural gradient works efficiently in learning. **Neural Computation**, MIT Press, Vancouver and Whistler, British Columbia, Canada, v. 10, n. 2, p. 251–276, 1998.

AMBIKASARAN, S.; FOREMAN-MACKEY, D.; GREENGARD, L.; HOGG, D. W.; O'NEIL, M. Fast direct methods for Gaussian processes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 38, n. 2, p. 252–265, 2016.

AŽMAN, K.; KOCIJAN, J. Dynamical systems identification using Gaussian process models with incorporated local models. **Engineering Applications of Artificial Intelligence**, v. 24, n. 2, p. 398–408, 2011.

BARBER, D. **Bayesian reasoning and machine learning**. Cambridge, UK: Cambridge University Press, 2012.

BAUER, M.; WILK, M. van der; RASMUSSEN, C. E. Understanding probabilistic sparse Gaussian process approximations. In: **Advances in Neural Information Processing Systems 29 (NIPS)**. Barcelona, Spain: JMLR.org, 2016. p. 1525–1533.

BERGER, B.; RAUSCHER, F. Robust Gaussian process modelling for engine calibration. **IFAC Proceedings Volumes**, Elsevier, v. 45, n. 2, p. 159–164, 2012.

BIJL, H. **Gaussian process regression techniques - with applications to wind turbines**. PhD Thesis — Delft University of Technology, 2016.

BIJL, H.; SCHÖN, T. B.; WINGERDEN, J.-W. van; VERHAEGEN, M. Online sparse Gaussian process training with input noise. **arXiv preprint arXiv:1601.08068**, 2016. Available on: <https://arxiv.org/abs/1601.08068>.

BILLINGS, S. A. **Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains**. Hoboken, NJ, USA: John Wiley & Sons, 2013.

BISHOP, C. M. **Pattern recognition and machine learning**. New York, NY, USA: Springer, 2006.

BLEI, D. M.; KUCUKELBIR, A.; MCAULIFFE, J. D. Variational inference: A review for statisticians. **Journal of the American Statistical Association**, Taylor & Francis, n. just-accepted, 2017.

BOTTEGAL, G.; ARAVKIN, A. Y.; HJALMARSSON, H.; PILLONETTO, G. Outlier robust system identification: a Bayesian kernel-based approach. **IFAC Proceedings Volumes**, Elsevier, v. 47, n. 3, p. 1073–1078, 2014.

BOTTOU, L. Online learning and stochastic approximations. **On-line learning in neural networks**, Cambridge University Press, v. 17, n. 9, p. 142, 1998.

BOTTOU, L. Stochastic learning. In: **Advanced Lectures on Machine Learning**. New York, NY, USA: Springer, 2004. p. 146–168.

BRAHIM-BELHOUARI, S.; BERMAK, A. Gaussian process for nonstationary time series prediction. **Computational Statistics & Data Analysis**, Elsevier, v. 47, n. 4, p. 705–712, 2004.

BUI, T. D.; HERNÁNDEZ-LOBATO, D.; LI, Y.; HERNÁNDEZ-LOBATO, J. M.; TURNER, R. E. Deep Gaussian processes for regression using approximate expectation propagation. In: **Proceedings of The 33rd International Conference on Machine Learning (ICML)**. New York City, NY, USA: JMLR.org, 2016.

BUI, T. D.; TURNER, R. E. Tree-structured Gaussian process approximations. In: **Advances in Neural Information Processing Systems 27 (NIPS)**. Montréal, Canada: NIPS Foundation, 2014. p. 2213–2221.

BUI, T. D.; TURNER, R. E. Stochastic variational inference for Gaussian process latent variable models using back constraints. In: **NIPS Workshop on Black Box Learning and Inference**. Montréal, Canada: NIPS Foundation, 2015.

CALANDRA, R.; PETERS, J.; RASMUSSEN, C. E.; DEISENROTH, M. P. Manifold Gaussian processes for regression. In: IEEE. **International Joint Conference on Neural Networks (IJCNN)**. Vancouver, Canada, 2016. p. 3338–3345.

CAMPOS, J.; LEWIS, F. L.; SELMIC, R. Backlash compensation with filtered prediction in discrete time nonlinear systems by dynamic inversion using neural networks. In: **Proceedings of the 39th IEEE Conference on Decision and Control (CDC)**. Sydney, Australia: IEEE, 2000. v. 4, p. 3534–3540.

CARLI, F. P.; CHIUSO, A.; PILLONETTO, G. Efficient algorithms for large scale linear system identification using stable spline estimators. In: **Proceedings of the 16th IFAC Symposium on System Identification (SYSID)**. Brussels, Belgium: Elsevier, 2012. v. 45, n. 16, p. 119–124.

CHATZIS, S. P.; DEMIRIS, Y. Echo state Gaussian process. **IEEE Transactions on Neural Networks**, IEEE, v. 22, n. 9, p. 1435–1445, 2011.

CHENG, Y.; WANG, Y.; CAMPS, O.; SZNAIER, M. The interplay between big data and sparsity in systems identification: Some lessons from machine learning. In: **Proceedings of the 17th IFAC Symposium on System Identification (SYSID)**. Beijing, China: Elsevier, 2015. v. 48, n. 28, p. 1285–1292.

CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014. Available on: <https://arxiv.org/abs/1406.1078>.

CHUESHOV, I. **Introduction to the theory of infinite-dimensional dissipative systems**. Canada: ACTA Scientific Publishing House, 2002.

COX, D. R. **Principles of statistical inference**. Cambridge, UK: Cambridge University Press, 2006.

CUTAJAR, K.; BONILLA, E. V.; MICHIARDI, P.; FILIPPONE, M. Practical learning of deep Gaussian processes via random fourier features. **arXiv preprint arXiv:1610.04386**, 2016. Available on: <https://arxiv.org/abs/1610.04386>.

DAI, Z.; DAMIANOU, A.; GONZÁLEZ, J.; LAWRENCE, N. Variational auto-encoded deep Gaussian processes. ICLR, San Juan Puerto Rico, 2016.

DAI, Z.; DAMIANOU, A.; HENSMAN, J.; LAWRENCE, N. Gaussian process models with parallelization and GPU acceleration. **arXiv preprint arXiv:1410.4984**, 2014. Available on: <https://arxiv.org/pdf/1410.4984>.

DAMIANOU, A. **Deep Gaussian processes and variational propagation of uncertainty**. PhD Thesis — University of Sheffield, 2015.

DAMIANOU, A.; LAWRENCE, N. Deep Gaussian processes. In: **Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Scottsdale, AZ, USA: JMLR.org, 2013. p. 207–215.

DAMIANOU, A.; LAWRENCE, N. Semi-described and semi-supervised learning with Gaussian processes. In: **31st Conference on Uncertainty in Artificial Intelligence (UAI)**. Amsterdam, Netherlands: AUAI Press, 2015.

DAMIANOU, A.; TITSIAS, M. K.; LAWRENCE, N. D. Variational Gaussian process dynamical systems. In: **Advances in Neural Information Processing Systems 24 (NIPS)**. Granada, Spain: NIPS Foundation, 2011. p. 2510–2518.

DAMIANOU, A. C.; TITSIAS, M. K.; LAWRENCE, N. D. Variational inference for latent variables and uncertain inputs in Gaussian processes. **Journal of Machine Learning Research (JMLR)**, v. 17, n. 42, p. 1–62, 2016.

DEISENROTH, M.; RASMUSSEN, C. E. PILCO: A model-based and data-efficient approach to policy search. In: **Proceedings of the 28th International Conference on Machine Learning (ICML)**. Bellevue, Washington, USA: JMLR.org, 2011. p. 465–472.

DEISENROTH, M. P.; NG, J. W. Distributed Gaussian processes. In: **Proceedings of the 32nd International Conference on Machine Learning (ICML)**. Lille, France: JMLR.org, 2015. p. 1481–1490.

DEZFOULI, A.; BONILLA, E. V. Scalable inference for Gaussian process models with black-box likelihoods. In: **Advances in Neural Information Processing Systems 28 (NIPS)**. Montréal, Canada: NIPS Foundation, 2015. p. 1414–1422.

DUVENAUD, D.; RIPPEL, O.; ADAMS, R.; GHAHRAMANI, Z. Avoiding pathologies in very deep networks. In: **Proceedings of the 17th Conference on Artificial Intelligence and Statistics (AISTATS)**. Reykjavik, Iceland: JMLR.org, 2014.

EK, C. H.; TORR, P. H.; LAWRENCE, N. D. Gaussian process latent variable models for human pose estimation. In: SPRINGER. **International Workshop on Machine Learning for Multimodal Interaction**. Brno, Czech Republic, 2007. p. 132–143.

ENGEL, Y.; MANNOR, S.; MEIR, R. The kernel recursive least-squares algorithm. **IEEE Transactions on Signal Processing**, IEEE, v. 52, n. 8, p. 2275–2285, 2004.

FERRIS, B.; FOX, D.; LAWRENCE, N. D. WiFi-SLAM using Gaussian process latent variable models. In: **International Joint Conference on Artificial Intelligence (IJCAI)**. Hyderabad, India: ACM, 2007. v. 7, n. 1, p. 2480–2485.

FLETCHER, R. **Practical methods of optimization**. Hoboken, NJ, USA: John Wiley & Sons, 2013.

FRIGOLA-ALCADE, R. **Bayesian Time Series Learning with Gaussian Processes**. PhD Thesis — University of Cambridge, 2015.

FRIGOLA-ALCADE, R.; CHEN, Y.; RASMUSSEN, C. Variational Gaussian process state-space models. In: **Advances in Neural Information Processing Systems 27 (NIPS)**. Cambridge, MA, USA: MIT Press, 2014. p. 3680–3688.

FRIGOLA-ALCADE, R.; LINDSTEN, F.; SCHÖN, T. B.; RASMUSSEN, C. E. Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In: **Advances in Neural Information Processing Systems 26 (NIPS)**. Lake Tahoe, Nevada, USA: NIPS Foundation, 2013. p. 3156–3164.

FRIGOLA-ALCADE, R.; RASMUSSEN, C. E. Integrated pre-processing for Bayesian nonlinear system identification with Gaussian processes. In: IEEE. **52nd IEEE Conference on Decision and Control (CDC)**. Firenze, Italy, 2013. p. 5371–5376.

GAL, Y.; WILK, M. van der; RASMUSSEN, C. E. Distributed variational inference in sparse Gaussian process regression and latent variable models. In: **Advances in Neural Information Processing Systems 27 (NIPS)**. Montréal, Canada: NIPS Foundation, 2014. p. 3257–3265.

GELMAN, A.; CARLIN, J. B.; STERN, H. S.; RUBIN, D. B. **Bayesian data analysis**. Boca Raton, FL, USA: Chapman & Hall, 2014.

GELMAN, A.; VEHTARI, A.; JYLÄNKI, P.; ROBERT, C.; CHOPIN, N.; CUNNINGHAM, J. P. Expectation propagation as a way of life. **arXiv preprint arXiv:1412.4869**, 2014. Available on: <https://arxiv.org/pdf/1412.4869>.

GIBBS, M. N.; MACKAY, D. J. Variational Gaussian process classifiers. **IEEE Transactions on Neural Networks**, IEEE, v. 11, n. 6, p. 1458–1464, 2000.

GIRARD, A. **Approximate methods for propagation of uncertainty with Gaussian process models**. PhD Thesis — University of Glasgow, 2004.

GIRARD, A.; RASMUSSEN, C.; QUIÑONERO-CANDELA, J.; MURRAY-SMITH, R. Multiple-step ahead prediction for non linear dynamic systems: A Gaussian process treatment with propagation of the uncertainty. In: **Advances in Neural Information Processing Systems 16 (NIPS)**. Cambridge, MA, USA: MIT Press, 2003. p. 529–536.

GIRARD, A.; RASMUSSEN, C. E.; QUIÑONERO-CANDELA, J.; MURRAY-SMITH, R. Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting. In: **Advances in Neural Information Processing Systems 15 (NIPS)**. Vancouver, Canada: MIT Press, 2002.

GLOROT, X.; BORDES, A.; BENGIO, Y. Deep sparse rectifier neural networks. In: **Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Ft. Lauderdale, Florida, USA: JMLR.org, 2011. v. 15, n. 106, p. 315–323.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Massachusetts, USA: MIT Press, 2016. Available on: <http://www.deeplearningbook.org>.

GREEN, P.; CROSS, E.; WORDEN, K. Bayesian system identification of dynamical systems using highly informative training data. **Mechanical Systems and Signal Processing**, Elsevier, v. 56, p. 109–122, 2015.

GREEN, P.; MASKELL, S. Estimating the parameters of dynamical systems from big data using sequential Monte Carlo samplers. **Mechanical Systems and Signal Processing**, Elsevier, v. 93, p. 379–396, 2017.

GREFF, K.; SRIVASTAVA, R. K.; KOUTNÍK, J.; STEUNEBRINK, B. R.; SCHMIDHUBER, J. LSTM: A search space odyssey. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, 2016.

GREGORCIC, G.; LIGHTBODY, G. Gaussian processes for modelling of dynamic non-linear systems. In: **Proceedings of the Irish Signals and Systems Conference (ISSC)**. Cork, Ireland: ISSC, 2002. p. 141–147.

GREGORČIČ, G.; LIGHTBODY, G. Nonlinear system identification: From multiple-model networks to Gaussian processes. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 21, n. 7, p. 1035–1055, 2008.

GUPTA, M.; GAO, J.; AGGARWAL, C. C.; HAN, J. Outlier detection for temporal data: A survey. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 26, n. 9, p. 2250–2267, 2014.

HAWKINS, D. M. **Identification of outliers**. USA: Springer, 1980.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: **Proceedings of the IEEE International Conference on Computer Vision (ICCV)**. Santiago, Chile: IEEE, 2015. p. 1026–1034.

HENSMAN, J.; DAMIANOU, A.; LAWRENCE, N. **Opening the way for deep Gaussian processes on massive data**. 2014. International Conference on Artificial Intelligence and Statistics (AISTATS), Late Breaking Poster.

HENSMAN, J.; DURRANDE, N.; SOLIN, A. Variational fourier features for Gaussian processes. **arXiv preprint arXiv:1611.06740**, 2016. Available on: <https://arxiv.org/abs/1611.06740>.

HENSMAN, J.; FUSI, N.; LAWRENCE, N. D. Gaussian processes for big data. In: **29th Conference on Uncertainty in Artificial Intelligence (UAI)**. Bellevue, Washington, USA: AUAI Press, 2013. p. 282–290.

HENSMAN, J.; LAWRENCE, N. D. Nested variational compression in deep Gaussian processes. **arXiv preprint arXiv:1412.1370**, 2014.

HENSMAN, J.; MATTHEWS, A. G. d. G.; GHAHRAMANI, Z. Scalable variational Gaussian process classification. In: **Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)**. San Diego, California, USA: JMLR.org, 2015.

HENTER, G. E.; FREAN, M. R.; KLEIJN, W. B. Gaussian process dynamical models for nonparametric speech representation and synthesis. In: IEEE. **IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. Kyoto, Japan, 2012. p. 4505–4508.

HERNÁNDEZ-LOBATO, D.; HERNÁNDEZ-LOBATO, J. M.; DUPONT, P. Robust multi-class Gaussian process classification. In: **Advances in Neural Information Processing Systems 24 (NIPS)**. Granada, Spain: NIPS Foundation, 2011. p. 280–288.

HJALMARSSON, H.; ROJAS, C. R.; RIVERA, D. E. System identification: A wiener-hammerstein benchmark. **Control Engineering Practice**, Elsevier, v. 20, n. 11, p. 1095–1096, 2012.

HOANG, T. N.; HOANG, Q. M.; LOW, B. K. H. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In: **Proceedings of the 32nd International Conference on Machine Learning (ICML)**. Lille, France: JMLR.org, 2015. p. 569–578.

HOANG, T. N.; HOANG, Q. M.; LOW, B. K. H. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In: **Proceedings of the 33rd International Conference on Machine Learning (ICML)**. New York City, NY, USA: JMLR.org, 2016. p. 382–391.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HOFFMAN, M. D.; BLEI, D. M.; WANG, C.; PAISLEY, J. W. Stochastic variational inference. **Journal of Machine Learning Research (JMLR)**, v. 14, n. 1, p. 1303–1347, 2013.

HUBER, P. J. **Robust statistics**. New York, NY, USA: Springer, 2011.

ISERMANN, R.; MÜNCHHOF, M. **Identification of dynamic systems**. Berlin, Alemanha: Springerverlag Berlin Heidelberg, 2011.

JAEGER, H. **The "echo state" approach to analysing and training recurrent neural networks - with an erratum note**. Bonn, Germany, 2001. v. 148, n. 34, 13 p. Technical report, German National Research Center for Information Technology GMD.

JAYNES, E. T. **Probability theory: The logic of science**. Cambridge, UK: Cambridge University Press, 2003.

JEFFREYS, H. **The theory of probability**. [S.l.]: OUP Oxford, 1998.

JIANG, Y.; SAXENA, A. Modeling high-dimensional humans for activity anticipation using Gaussian process latent CRFs. In: **Robotics: Science and Systems**. Berkeley, California, USA: MIT Press, 2014. p. 1–8.

JORDAN, M. I. **An introduction to probabilistic graphical models**. 2003. Unpublished.

JORDAN, M. I.; GHAHRAMANI, Z.; JAAKKOLA, T. S.; SAUL, L. K. An introduction to variational methods for graphical models. **Machine Learning**, Springer, v. 37, n. 2, p. 183–233, 1999.

JURICIC, D.; KOCIJAN, J. Fault detection based on Gaussian process model. In: ARGESIM. **Proceedings of the 5th Vienna Symposium on Mathematical Modeling (MathMod)**. Vienna, Austria, 2006.

JYLÄNKI, P.; VANHATALO, J.; VEHTARI, A. Robust Gaussian process regression with a student-t likelihood. **Journal of Machine Learning Research (JMLR)**, JMLR.org, v. 12, n. Nov, p. 3227–3257, 2011.

KALMAN, R. E. *et al.* A new approach to linear filtering and prediction problems. **Journal of Basic Engineering**, ASME, v. 82, n. 1, p. 35–45, 1960.

KANTAS, N.; DOUCET, A.; SINGH, S. S.; MACIEJOWSKI, J.; CHOPIN, N. *et al.* On particle methods for parameter estimation in state-space models. **Statistical Science**, IMS, Bethesda, MD, USA, v. 30, n. 3, p. 328–351, 2015.

KAUFMAN, L.; ROUSSEEUW, P. J. Partitioning around medoids (program PAM). **Finding groups in data: an introduction to cluster analysis**, John Wiley & Sons, Hoboken, NJ, USA, p. 68–125, 1990.

KIM, H.-C.; GHAHRAMANI, Z. Outlier robust Gaussian process classification. **Structural, Syntactic, and Statistical Pattern Recognition**, Springer, p. 896–905, 2008.

KIM, Y.; MALLICK, R.; BHOWMICK, S.; CHEN, B.-L. Nonlinear system identification of large-scale smart pavement systems. **Expert Systems with Applications**, Elsevier, v. 40, n. 9, p. 3551–3560, 2013.

KING, N. J.; LAWRENCE, N. D. Fast variational inference for Gaussian process models through KL-correction. In: **Proceedings of the 17th European Conference on Machine Learning (ECML)**. Berlin, Germany: Springer, 2006. p. 270–281.

KINGMA, D.; BA, J. Adam: A method for stochastic optimization. In: **International Conference on Learning Representations (ICLR)**. San Diego, California, USA: ICLR, 2015.

KINGMA, D. P.; WELLING, M. Auto-Encoding Variational Bayes. In: **International Conference on Learning Representations (ICLR)**. Banff, Canada: ICLR, 2014.

KLENSKE, E. D.; ZEILINGER, M. N.; SCHÖLKOPF, B.; HENNIG, P. Gaussian process-based predictive control for periodic error correction. **IEEE Transactions on Control Systems Technology**, IEEE, v. 24, n. 1, p. 110–121, 2016.

KO, J.; KLEIN, D. J.; FOX, D.; HAEHNEL, D. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In: IEEE. **IEEE International Conference on Robotics and Automation (ICRA)**. Rome, Italy, 2007. p. 742–747.

KOCIJAN, J. **Modelling and Control of Dynamic Systems Using Gaussian Process Models**. New York, NY, USA: Springer, 2016.

KOCIJAN, J.; GIRARD, A.; BANKO, B.; MURRAY-SMITH, R. Dynamic systems identification with Gaussian processes. **Mathematical and Computer Modelling of Dynamical Systems**, v. 11, n. 4, p. 411–424, 2005.

KOCIJAN, J.; MURRAY-SMITH, R.; RASMUSSEN, C. E.; GIRARD, A. Gaussian process model based predictive control. In: IEEE. **Proceedings of the American Control Conference (ACC)**. Boston, Massachusetts, 2004. v. 3, p. 2214–2219.

KOCIJAN, J.; PETELIN, D. Output-error model training for Gaussian process models. In: **International Conference on Adaptive and Natural Computing Algorithms (ICANNGA)**. Ljubljana, Slovenia: Springer, 2011. p. 312–321.

KOLLER, D.; FRIEDMAN, N. **Probabilistic graphical models: principles and techniques**. Cambridge, MA, USA: MIT Press, 2009.

KORIYAMA, T.; NOSE, T.; KOBAYASHI, T. Statistical parametric speech synthesis based on Gaussian process regression. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 8, n. 2, p. 173–183, 2014.

KUSS, M. **Gaussian process models for robust regression, classification, and reinforcement learning**. PhD Thesis — TU Darmstadt, 2006.

KUSS, M.; PFINGSTEN, T.; CSATÓ, L.; RASMUSSEN, C. E. **Approximate inference for robust Gaussian process regression**. Tubingen, Germany, 2005. v. 136. Technical report, Max Planck Institute for Biological Cybernetics.

KUSS, M.; RASMUSSEN, C. E. Assessing approximate inference for binary Gaussian process classification. **Journal of Machine Learning Research (JMLR)**, v. 6, n. Oct, p. 1679–1704, 2005.

LAWRENCE, N. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. **Journal of Machine Learning Research (JMLR)**, v. 6, p. 1783–1816, 2005.

LAWRENCE, N. D. Gaussian process latent variable models for visualisation of high dimensional data. In: **Advances in Neural Information Processing Systems 17 (NIPS)**. Vancouver and Whistler, British Columbia, Canada: MIT Press, 2004. p. 329–336.

LAWRENCE, N. D.; MOORE, A. J. Hierarchical Gaussian process latent variable models. In: **Proceedings of the 24th International Conference on Machine Learning (ICML)**. Corvallis, OR, USA: ACM, 2007. p. 481–488.

LAWRENCE, N. D.; QUIÑONERO-CANDELA, J. Local distance preservation in the GP-LVM through back constraints. In: ACM. **Proceedings of the 23rd International Conference on Machine Learning (ICML)**. Pittsburgh, USA, 2006. p. 513–520.

LÁZARO-GREDILLA, M. Bayesian warped Gaussian processes. In: **Advances in Neural Information Processing Systems 25 (NIPS)**. Lake Tahoe, Nevada, USA: NIPS Foundation, 2012. p. 1619–1627.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group, London, UK, v. 521, n. 7553, p. 436–444, 2015.

LI, X.-M.; OUYANG, J.-H. Tuning the learning rate for stochastic variational inference. **Journal of Computer Science and Technology**, Springer Science & Business Media, v. 31, n. 2, p. 428, 2016.

LIKAR, B.; KOCIJAN, J. Predictive control of a gas-liquid separation plant based on a Gaussian process model. **Computers & Chemical Engineering**, Elsevier, v. 31, n. 3, p. 142–152, 2007.

LIU, W.; POKHAREL, P. P.; PRINCIPE, J. C. The kernel least-mean-square algorithm. **IEEE Transactions on Signal Processing**, IEEE, v. 56, n. 2, p. 543–554, 2008.

LJUNG, L. **System Identification - Theory for the User**. USA: Wiley Online Library, 1999.

MACKEY, M. C.; GLASS, L. *et al.* Oscillation and chaos in physiological control systems. **Science**, AAAS, Washington, D.C., USA, v. 197, n. 4300, p. 287–289, 1977.

MAECHLER, M.; ROUSSEEUW, P.; STRUYF, A.; HUBERT, M.; HORNIK, K. **cluster: Cluster Analysis Basics and Extensions**. Zürich, 2016. R package version 2.0.4.

MAJHI, B.; PANDA, G. Robust identification of nonlinear complex systems using low complexity ANN and particle swarm optimization technique. **Expert Systems with Applications**, v. 38, n. 1, p. 321–333, 2011.

MARCONATO, A.; SJÖBERG, J.; SUYKENS, J.; SCHOUKENS, J. Identification of the silverbox benchmark using nonlinear state-space models. In: **Proceedings of the 16th IFAC Symposium on System Identification (SYSID)**. Brussels, Belgium: Elsevier, 2012. v. 45, n. 16, p. 632–637.

MATHERON, G. The intrinsic random functions and their applications. **Advances in Applied Probability**, JSTOR, p. 439–468, 1973.

MATLAB. **version 8.1.0.604 (R2013a)**. Natick, Massachusetts: The MathWorks Inc., 2013.

MATTHEWS, A. G. d. G.; HENSMAN, J.; TURNER, R.; GHAHRAMANI, Z. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In: **Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Cadiz, Spain: JMLR.org, 2016. p. 231–239.

MATTOS, C. L. C.; DAI, Z.; DAMIANOU, A.; FORTH, J.; BARRETO, G. A.; LAWRENCE, N. D. Recurrent Gaussian processes. In: **International Conference on Learning Representations (ICLR)**. San Juan Puerto Rico: ICLR, 2016.

MATTOS, C. L. C.; DAI, Z.; DAMIANOU, A.; BARRETO, G. A.; LAWRENCE, N. D. Deep recurrent gaussian processes for outlier-robust system identification. **Journal of Process Control**, Elsevier, 2017.

MATTOS, C. L. C.; DAMIANOU, A.; BARRETO, G. A.; LAWRENCE, N. D. Latent autoregressive Gaussian process models for robust system identification. In: **Proceedings of the 11th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS-CAB)**. Trondheim, Norway: IFAC Publisher, 2016. p. 1121–1126.

MATTOS, C. L. C.; SANTOS, J. D. A.; BARRETO, G. A. An empirical evaluation of robust Gaussian process models for system identification. In: **Intelligent Data Engineering and Automated Learning (IDEAL)**. Wroclaw, Poland: Springer, 2015. p. 172–180.

MCHUTCHON, A. **Nonlinear modelling and control using Gaussian processes**. PhD Thesis — University of Cambridge, 2014.

MCHUTCHON, A.; RASMUSSEN, C. E. Gaussian process training with input noise. In: **Advances in Neural Information Processing Systems 24 (NIPS)**. Granada, Spain: NIPS Foundation, 2011. p. 1341–1349.

MILANESE, M.; NORTON, J.; PIET-LAHANIER, H.; WALTER, É. **Bounding approaches to system identification**. New York, NY, USA: Springer Science & Business Media, 2013.

MINKA, T. P. Expectation propagation for approximate Bayesian inference. In: **Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)**. Seattle, WA, USA: Morgan Kaufmann, 2001. p. 362–369.

MOOR, B. L. R. D. **DaISy: Database for the Identification of Systems**. 2016. Available on: <http://homes.esat.kuleuven.be/~smc/daisy/>. Accessed July 2016.

MURPHY, K. P. **Machine learning: a probabilistic perspective**. Cambridge, MA, USA: MIT Press, 2012.

MURRAY-SMITH, R.; GIRARD, A. Gaussian process priors with ARMA noise models. In: **Irish Signals and Systems Conference (ISSC)**. Maynooth, Ireland: National University of Ireland, 2001. p. 147–152.

MURRAY-SMITH, R.; JOHANSEN, T. A.; SHORTEN, R. On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In: **European Control Conference (ECC)**. Karlsruhe, Germany: Springer, 1999.

NAISH-GUZMAN, A.; HOLDEN, S. Robust regression with twinned Gaussian processes. In: **Advances in Neural Information Processing Systems 21 (NIPS)**. Vancouver, Canada: MIT Press, 2008. p. 1065–1072.

NARENDRA, K. S.; LI, S.-M. Neural networks in control systems. **Mathematical Perspectives on Neural Networks**, Psychology Press, p. 347–394, 1996.

NARENDRA, K. S.; PARTHASARATHY, K. Identification and control of dynamical systems using neural networks. **IEEE Transactions on Neural Networks**, v. 1, n. 1, p. 4–27, 1990.

NEAL, R. M. **Bayesian learning for neural networks**. PhD Thesis — University of Toronto, 1994.

NEAL, R. M. **Monte Carlo implementation of Gaussian process models for Bayesian regression and classification**. Toronto, Canada, 1997. Technical report, University of Toronto, Dept. of Statistics.

NELLES, O. **Nonlinear system identification: from classical approaches to neural networks and fuzzy models**. Berlin, Germany: Springer Science & Business Media, 2013.

NEO, K. K. S. **Non-linear dynamics identification using Gaussian process prior models within a Bayesian context**. Tese (Doutorado) — National University of Ireland Maynooth, 2008.

NGUYEN, T.; BONILLA, E. Fast allocation of Gaussian process experts. In: **Proceedings of the 31st International Conference on Machine Learning (ICML)**. Beijing, China: JMLR.org, 2014. p. 145–153.

NICKSON, T.; GUNTER, T.; LLOYD, C.; OSBORNE, M. A.; ROBERTS, S. Blitzkriging: Kronecker-structured stochastic Gaussian processes. **arXiv preprint arXiv:1510.07965**, 2015. Available on: <https://arxiv.org/pdf/1510.07965>.

OPPER, M.; WINTHER, O. Gaussian processes for classification: Mean-field algorithms. **Neural Computation**, MIT Press, v. 12, n. 11, p. 2655–2684, 2000.

OSBORNE, M. A.; GARNETT, R.; SWERSKY, K.; FREITAS, N. D. Prediction and fault detection of environmental signals with uncharacterised faults. In: **Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Toronto, Ontario, Canada: JMLR.org, 2012.

OSBORNE, M. A.; ROBERTS, S. J.; ROGERS, A.; RAMCHURN, S. D.; JENNINGS, N. R. Towards real-time information processing of sensor network data using computationally efficient multi-output Gaussian processes. In: IEEE COMPUTER SOCIETY. **Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN)**. St. Louis, Missouri, USA, 2008. p. 109–120.

PARISI, G. **Statistical field theory**. USA: Addison-Wesley, 1988.

PASCANU, R.; GULCEHRE, C.; CHO, K.; BENGIO, Y. How to construct deep recurrent neural networks. In: **International Conference on Learning Representations (ICLR)**. Scottsdale, AZ, USA: ICLR, 2014.

PEARSON, R. K. Outliers in process modeling and identification. **IEEE Transactions on Control Systems Technology**, IEEE, v. 10, n. 1, p. 55–63, 2002.

PÉREZ-CRUZ, F.; VAERENBERGH, S. V.; MURILLO-FUENTES, J. J.; LÁZARO-GREDILLA, M.; SANTAMARIA, I. Gaussian processes for nonlinear signal processing: An overview of recent advances. **IEEE Signal Processing Magazine**, IEEE, v. 30, n. 4, p. 40–50, 2013.

PETERKA, V. Bayesian approach to system identification. **Trends and Progress in System Identification**, Pergamon Press Oxford, v. 1, p. 239–304, 1981.

PILLONETTO, G. The interplay between system identification and machine learning. **arXiv preprint arXiv:1612.09158**, 2016. Available on: <https://arxiv.org/pdf/1612.09158>.

PILLONETTO, G.; DINUZZO, F.; CHEN, T.; NICOLAO, G. D.; LJUNG, L. Kernel methods in system identification, machine learning and function estimation: A survey. **Automatica**, Elsevier, v. 50, n. 3, p. 657–682, 2014.

QUIÑONERO-CANDELA, J.; GIRARD, A. **Prediction at an uncertain input for Gaussian processes and relevance vector machines - application to multiple-step ahead time-series forecasting**. Kongens Lyngby, Denmark, 2002. Technical report, Technical University of Denmark, Dept. of Informatics and Mathematical Modelling.

QUIÑONERO-CANDELA, J.; GIRARD, A.; LARSEN, J.; RASMUSSEN, C. E. Propagation of uncertainty in Bayesian kernel models-application to multiple-step ahead forecasting. In: **Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)**. Hong Kong, Hong Kong: IEEE, 2003. v. 2, p. II–701.

QUIÑONERO-CANDELA, J.; RASMUSSEN, C. E. A unifying view of sparse approximate Gaussian process regression. **The Journal of Machine Learning Research (JMLR)**, v. 6, p. 1939–1959, 2005.

R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2017. Available on: <https://www.R-project.org/>.

RAISSI, M. Parametric Gaussian process regression for big data. **arXiv preprint arXiv:1704.03144**, 2017. Available on: <https://arxiv.org/pdf/1704.03144>.

RANGANATH, R.; WANG, C.; DAVID, B.; XING, E. An adaptive learning rate for stochastic variational inference. In: **Proceedings of the 30th International Conference on Machine Learning (ICML)**. Atlanta, USA: JMLR.org, 2013. p. 298–306.

RANJAN, R.; HUANG, B.; FATEHI, A. Robust Gaussian process modeling using em algorithm. **Journal of Process Control**, Elsevier, v. 42, p. 125–136, 2016.

RASMUSSEN, C.; WILLIAMS, C. **Gaussian Processes for Machine Learning**. 1. ed. Cambridge, MA, USA: MIT Press, 2006.

RASMUSSEN, C. E. **Evaluation of Gaussian processes and other methods for non-linear regression**. PhD Thesis — University of Toronto, Toronto, Canada, 1996.

RAUCH, H. E.; STRIEBEL, C.; TUNG, F. Maximum likelihood estimates of linear dynamic systems. **American Institute of Aeronautics and Astronautics (AIAA) Journal**, AIAA, v. 3, n. 8, p. 1445–1450, 1965.

REECE, S.; ROBERTS, S. An introduction to Gaussian processes for the Kalman filter expert. In: IEEE. **13th Conference on Information Fusion (FUSION)**. Edinburgh, UK, 2010. p. 1–9.

REZENDE, D. J.; MOHAMED, S.; WIERSTRA, D. Stochastic backpropagation and approximate inference in deep generative models. In: **Proceedings of the 31st International Conference on Machine Learning (ICML)**. Beijing, China: JMLR.org, 2014.

ROBBINS, H.; MONRO, S. A stochastic approximation method. **The Annals of Mathematical Statistics**, JSTOR, p. 400–407, 1951.

ROCHA, F. H. M. D.; GRASSI, V.; GUIZILINI, V. C.; RAMOS, F. Model predictive control of a heavy-duty truck based on Gaussian process. In: IEEE. **XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)**. Recife, Pernambuco, Brazil, 2016. p. 97–102.

ROTTMANN, A.; BURGARD, W. Learning non-stationary system dynamics online using Gaussian processes. **Pattern Recognition**, Springer, v. 6373, p. 192–201, 2010.

ROUSSEEUW, P. J.; LEROY, A. M. **Robust regression and outlier detection**. Hoboken, NJ, USA: John Wiley & Sons, 2005.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In: **Parallel Distributed Processing: Explorations in the Microstructure of Cognition**. Cambridge, Massachusetts, USA: MIT Press, 1986. v. 1, p. 318–362.

SALIMBENI, H.; DEISENROTH, M. Doubly stochastic variational inference for deep Gaussian processes. **arXiv preprint arXiv:1705.08933**, 2017. Available on: <https://arxiv.org/abs/1705.08933>.

SANTOS, J. D. A.; BARRETO, G. A. A regularized estimation framework for online sparse LSSVR models. **Neurocomputing**, Elsevier, v. 238, p. 114–125, 2017.

SÄRKKÄ; SOLIN, A.; HARTIKAINEN, J. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. **IEEE Signal Processing Magazine**, IEEE, v. 30, n. 4, p. 51–61, 2013.

SÄRKKÄ, S. **Bayesian filtering and smoothing**. Cambridge, UK: Cambridge University Press, 2013.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural Networks**, Pergamon, v. 61, p. 85–117, 2015.

SCHÖN, T. B.; LINDSTEN, F.; DAHLIN, J.; WÅGBERG, J.; NAESSETH, C. A.; SVENSSON, A.; DAI, L. Sequential monte carlo methods for system identification. In: **Proceedings of the 17th IFAC Symposium on System Identification (SYSID)**. Beijing, China: Elsevier, 2015. v. 48, n. 28, p. 775–786.

SCHOUKENS, J.; NEMETH, J. G.; CRAMA, P.; ROLAIN, Y.; PINTELON, R. Fast approximate identification of nonlinear systems. **Automatica**, Elsevier, v. 39, n. 7, p. 1267–1274, 2003.

SCHOUKENS, J.; SUYKENS, J.; LJUNG, L. Wiener-hammerstein benchmark. In: **Proceedings of the 15th IFAC Symposium on System Identification (SYSID)**. Saint-Malo, France: Elsevier, 2009.

SCHOUKENS, M.; MATTSON, P.; WIGREN, T.; NOËL, J. Cascaded tanks benchmark combining soft and hard nonlinearities. 2015. Available on: <http://homepages.vub.ac.be/~mschouke/benchmark2016.html>.

SCHOUKENS, M.; SCHEIWE, F. G. Modeling nonlinear systems using a Volterra feedback model. In: **Workshop on Nonlinear System Identification Benchmarks**. Brussels, Belgium: Vrije Universiteit Brussel, 2016.

SCHWARZ, H. R. **Finite element methods**. USA: Academic Press, 1988.

SHAWE-TAYLOR, J.; CRISTIANINI, N. **Kernel methods for pattern analysis**. Cambridge, UK: Cambridge University Press, 2004.

SJÖBERG, J.; ZHANG, Q.; LJUNG, L.; BENVENISTE, A.; DELYON, B.; GLORENNEC, P.-Y.; HJALMARSSON, H.; JUDITSKY, A. Nonlinear black-box modeling in system identification: a unified overview. **Automatica**, Elsevier, v. 31, n. 12, p. 1691–1724, 1995.

SMOLA, A. J.; SCHÖLKOPF, B. **Learning with kernels**. Cambridge, MA, USA: MIT Press, 2002.

SNELSON, E.; RASMUSSEN, C. E.; GHAHRAMANI, Z. Warped Gaussian processes. **Advances in Neural Information Processing Systems 17 (NIPS)**, MIT Press, Vancouver and Whistler, British Columbia, Canada, v. 16, p. 337–344, 2004.

SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical Bayesian optimization of machine learning algorithms. In: **Advances in Neural Information Processing Systems 25 (NIPS)**. Lake Tahoe, Nevada, USA: NIPS Foundation, 2012. p. 2951–2959.

SOLAK, E.; MURRAY-SMITH, R.; LEITHEAD, W. E.; LEITH, D. J.; RASMUSSEN, C. E. Derivative observations in Gaussian process models of dynamic systems. **Advances in Neural Information Processing Systems 16 (NIPS)**, MIT Press, Cambridge, MA, USA, v. 16, 2003.

SOLLICH, P. Bayesian methods for support vector machines: Evidence and predictive class probabilities. **Machine Learning**, Springer, v. 46, n. 1, p. 21–52, 2002.

STEGLE, O.; FALLERT, S. V.; MACKAY, D. J.; BRAGE, S. Gaussian process robust regression for noisy heart rate data. **IEEE Transactions on Biomedical Engineering**, IEEE, v. 55, n. 9, p. 2143–2151, 2008.

STEIN, M. L. **Interpolation of spatial data: some theory for kriging**. Berlin, Germany: Springer Science & Business Media, 2012.

SVENSSON, A. **Learning probabilistic models of dynamical phenomena using particle filters**. PhD Thesis — Uppsala University, 2016.

SVENSSON, A.; DAHLIN, J.; SCHÖN, T. B. Marginalizing Gaussian process hyperparameters using sequential Monte Carlo. In: IEEE. **IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)**. Cancun, Mexico, 2015. p. 477–480.

SVENSSON, A.; SCHÖN, T. B. A flexible state–space model for learning nonlinear dynamical systems. **Automatica**, Elsevier, v. 80, p. 189–199, 2017.

SVENSSON, A.; SOLIN, A.; SÄRKKÄ, S.; SCHÖN, T. B. Computationally efficient Bayesian learning of Gaussian process state space models. In: **Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Cadiz, Spain: JMLR.org, 2016. p. 213–221.

TALBI, E.-G. **Metaheuristics: from design to implementation**. Hoboken, NJ, USA: John Wiley & Sons, 2009.

The MathWorks Inc. **Nonlinear Modeling of a Magneto-Rheological Fluid Damper**. 2016. Available on: <http://www.mathworks.com/help/ident/examples/nonlinear-modeling-of-a-magneto-rheological-fluid-damper.html>. Accessed September 2016.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. **arXiv e-prints**, abs/1605.02688, 2016. Available on: <http://arxiv.org/abs/1605.02688>.

THOMPSON, K. R. **Implementation of Gaussian process models for non-linear system identification**. PhD Thesis — University of Glasgow, 2009.

TIPPING, M. E.; LAWRENCE, N. D. Variational inference for student-t models: Robust Bayesian interpolation and generalised component analysis. **Neurocomputing**, v. 69, n. 1, p. 123–141, 2005.

TITSIAS, M. K. Variational learning of inducing variables in sparse Gaussian processes. In: **Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Clearwater Beach, FL, USA: JMLR.org, 2009. p. 567–574.

TITSIAS, M. K. **Variational model selection for sparse Gaussian process regression**. Manchester, UK, 2009. Technical report, University of Manchester, School of Computer Science.

TITSIAS, M. K.; LAWRENCE, N. D. Bayesian Gaussian process latent variable model. In: **Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Sardinia, Italy: JMLR.org, 2010. p. 844–851.

TURNER, R. D. **Gaussian Processes for state space models and change point detection**. PhD Thesis — University of Cambridge, 2012.

TURNER, R. D.; DEISENROTH, M. P.; RASMUSSEN, C. E. State-space inference and learning with Gaussian processes. In: **Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Sardinia, Italy: JMLR.org, 2010. p. 868–875.

TURNER, R. E.; SAHANI, M. Two problems with variational expectation maximisation for time-series models. In: **Workshop on Inference and Estimation in Probabilistic Time-Series Models**. Cambridge, UK: Isaac Newton Institute for Mathematical Sciences, 2008. v. 2, n. 3.

VAERENBERGH, S. V.; FERNANDEZ-BES, J.; ELVIRA, V. On the relationship between online Gaussian process regression and kernel least mean squares algorithms. In: IEEE. **26th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)**. Vietri sul Mare, Salerno, Italy, 2016. p. 1–6.

VAERENBERGH, S. V.; LÁZARO-GREDILLA, M.; SANTAMARÍA, I. Kernel recursive least-squares tracker for time-varying regression. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 23, n. 8, p. 1313–1326, 2012.

VINOGRADSKA, J.; BISCHOFF, B.; NGUYEN-TUONG, D.; ROMER, A.; SCHMIDT, H.; PETERS, J. Stability of controllers for Gaussian process forward models. In: **Proceedings of the 33rd International Conference on Machine Learning (ICML)**. New York City, NY, USA: JMLR.org, 2016. p. 545–554.

WAINWRIGHT, M. J.; JORDAN, M. I. *et al.* Graphical models, exponential families, and variational inference. **Foundations and Trends in Machine Learning**, Now Publishers, Inc., Netherlands, v. 1, n. 1–2, p. 1–305, 2008.

WANG, J.; HERTZMANN, A.; BLEI, D. M. Gaussian process dynamical models. In: **Advances in Neural Information Processing Systems 18 (NIPS)**. Vancouver, British Columbia, Canada: MIT Press, 2005. p. 1441–1448.

WANG, J.; SANO, A.; CHEN, T.; HUANG, B. Identification of hammerstein systems without explicit parameterisation of non-linearity. **International Journal of Control**, Taylor & Francis, Abingdon, UK, v. 82, n. 5, p. 937–952, 2009.

WANG, J. M.; FLEET, D. J.; HERTZMANN, A. Gaussian process dynamical models for human motion. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 30, n. 2, p. 283–298, 2008.

WANG, Y.; BRUBAKER, M.; CHAIB-DRAA, B.; URTASUN, R. Sequential inference for deep Gaussian process. In: **Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Cadiz, Spain: JMLR.org, 2016. v. 52, p. 694–703.

WIGREN, T. **Input-output data sets for development and benchmarking in nonlinear identification**. Uppsala, Sweden, 2010. v. 20. Technical report, Uppsala University, Dept. of Information Technology.

WIGREN, T.; SCHOUKENS, J. **Data for benchmarking in nonlinear system identification**. Uppsala, Sweden, 2013. v. 6. Technical report, Uppsala University, Dept. of Information Technology.

WILLIAMS, C. K. Computation with infinite neural networks. **Neural Computation**, MIT Press, Cambridge, MA, USA, v. 10, n. 5, p. 1203–1216, 1998.

WILLIAMS, C. K.; BARBER, D. Bayesian classification with Gaussian processes. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 20, n. 12, p. 1342–1351, 1998.

WILLIAMS, C. K.; RASMUSSEN, C. E. Gaussian processes for regression. In: MIT PRESS. **Advances in Neural Information Processing Systems 9 (NIPS)**. Cambridge, MA, USA, 1996. p. 514–520.

WILLIAMS, R. J.; ZIPSER, D. Gradient-based learning algorithms for recurrent networks and their computational complexity. **Back-propagation: Theory, architectures and applications**, Lawrence Erlbaum Publishers, Hillsdale, New Jersey, USA, p. 433–486, 1995.

WILSON, A.; NICKISCH, H. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In: **Proceedings of the 32nd International Conference on Machine Learning (ICML)**. Lille, France: JMLR.org, 2015. p. 1775–1784.

WILSON, A. G.; DANN, C.; NICKISCH, H. Thoughts on massively scalable Gaussian processes. **arXiv preprint arXiv:1511.01870**, 2015. Available on: <https://arxiv.org/pdf/1511.01870>.

WILSON, A. G.; HU, Z.; SALAKHUTDINOV, R.; XING, E. P. Deep kernel learning. In: **Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)**. Cadiz, Spain: JMLR.org, 2016. p. 370–378.

WILSON, A. G.; HU, Z.; SALAKHUTDINOV, R. R.; XING, E. P. Stochastic variational deep kernel learning. In: **Advances in Neural Information Processing Systems 29 (NIPS)**. Barcelona, Spain: NIPS Foundation, 2016. p. 2586–2594.

WORDEN, K.; MANSON, G.; CROSS, E. J. On Gaussian process NARX models and their higher-order frequency response functions. **Solving Computationally Expensive Engineering Problems**, Springer, p. 315–335, 2014.

ZHU, Y. **Multivariable system identification for process control**. Netherlands: Elsevier, 2001.

APPENDIX A – MATHEMATICAL DETAILS AND DERIVATIONS

"Go down deep enough into anything and you will find mathematics."
(Dean Schlicter)

This appendix collects mathematical details of the modeling approaches presented in the main text of the thesis, following the same notation used so far.

A.1 Variational Lower Bound Statistics

Here we detail the statistics defined in Eq. (3.27), which are related to the ones presented in the original paper on Bayesian GP-LVM (TITSIAS; LAWRENCE, 2010).

First, we define the notation of the following variational distribution based on Eq. (3.14):

$$q\left(\hat{\boldsymbol{x}}_i^{(h)}\right) = \mathcal{N}\left(\hat{\boldsymbol{x}}_i^{(h)} \middle| \boldsymbol{\mu}_i^{(h)}, \boldsymbol{\Sigma}_i^{(h)}\right), \quad 1 \leq i \leq N-L, \;\; 1 \leq h \leq H+1, \tag{A.1}$$

where $\hat{\boldsymbol{x}}_i^{(h)} \in \mathbb{R}^{D_h}$, $\boldsymbol{\mu}_i^{(h)} \in \mathbb{R}^{D_h}$, $\boldsymbol{\Sigma}_i^{(h)} \in \mathbb{R}^{D_h \times D_h}$ and

$$D_h = \begin{cases} L + L_u, & \text{if } h = 1, \\ 2L, & \text{if } 1 < h \leq H, \\ L, & \text{if } h = H+1. \end{cases} \tag{A.2}$$

The moments themselves are given by

$$\boldsymbol{\mu}_i^{(h)} = \begin{cases} \left[\mu_{i-1}^{(1)}, \cdots, \mu_{i-L}^{(1)}, u_{i-1}, \cdots, u_{i-L_u}\right]^\top, & \text{if } h = 1, \\ \left[\mu_{i-1}^{(h)}, \cdots, \mu_{i-L}^{(h)}, \mu_i^{(h-1)}, \cdots, \mu_{i-L+1}^{(h-1)}\right]^\top, & \text{if } 1 < h \leq H, \\ \left[\mu_i^{(H)}, \cdots, \mu_{i-L+1}^{(H)}\right]^\top, & \text{if } h = H+1, \end{cases} \tag{A.3}$$

$$\boldsymbol{\Sigma}_i^{(h)} = \begin{cases} \mathrm{diag}\left(\left[\lambda_{i-1}^{(1)}, \cdots, \lambda_{i-L}^{(1)}, \mathbf{0}_{L_u}\right]\right), & \text{if } h = 1, \\ \mathrm{diag}\left(\left[\lambda_{i-1}^{(h)}, \cdots, \lambda_{i-L}^{(h)}, \lambda_i^{(h-1)}, \cdots, \lambda_{i-L+1}^{(h-1)}\right]\right), & \text{if } 1 < h \leq H, \\ \mathrm{diag}\left(\left[\lambda_i^{(H)}, \cdots, \lambda_{i-L+1}^{H}\right]\right), & \text{if } h = H+1, \end{cases} \tag{A.4}$$

where the function $\mathbf{diag}(\cdot)$ builds a diagonal matrix from its argument. Note that $\mathbf{0}_{L_u}$ is a vector of L_u zeros, related to the variances of the external inputs, which are actually considered deterministic. Conveniently, if we have the uncertainty information about those

variables, e.g., if they come from a Bayesian controller, we can simply replace this vector by the vector of known variances, given that they are (at least approximately) Gaussian distributed, without changing the RGP/REVARB framework. If that approach is pursued it is equivalent to approximately marginalize the inputs u_i, similar to the marginalization of the dynamical latent variables $x_i^{(1)}$ of the first layer.

We can now detail the statistics in Eq. (3.27). We will drop the indexes h that refer to the layer number to reduce clutter in the notation, but we emphasize that the computations are separately performed for each layer. We consider that the exponentiated quadratic covariance function was chosen.

The statistic $\Psi_0 = \mathrm{Tr}\left(\left\langle \boldsymbol{K}_f \right\rangle_{q(\boldsymbol{x})}\right) \in \mathbb{R}$ is given by

$$\begin{aligned}\Psi_0 &= \sum_{i=1}^{N-L} \int_{\hat{\boldsymbol{x}}_i} k\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_i\right) \mathcal{N}\left(\hat{\boldsymbol{x}}_i | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \\ &= (N-L)\sigma_f^2. \end{aligned} \tag{A.5}$$

The computation of each element of $\boldsymbol{\Psi}_1 = \left\langle \boldsymbol{K}_{fz} \right\rangle_{q(\boldsymbol{x})} \in \mathbb{R}^{(N-L)\times M}$ is performed as follows:

$$\begin{aligned}[\boldsymbol{\Psi}_1]_{ij} &= \int_{\hat{\boldsymbol{x}}_i} k\left(\hat{\boldsymbol{x}}_i, \boldsymbol{\zeta}_j\right) \mathcal{N}\left(\hat{\boldsymbol{x}}_i | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \\ &= \sigma_f^2 \prod_{d=1}^{D} \frac{\exp\left(-\frac{1}{2}\frac{w_d(x_{id}-\zeta_{jd})^2}{w_d[\boldsymbol{\Sigma}_i]_{dd}+1}\right)}{\left(w_d[\boldsymbol{\Sigma}_i]_{dd}+1\right)^{\frac{1}{2}}}. \end{aligned} \tag{A.6}$$

The statistic $\boldsymbol{\Psi}_2 = \left\langle \left(\boldsymbol{K}_{fz}\right)^\top \boldsymbol{K}_{fz} \right\rangle_{q(\boldsymbol{x})} \in \mathbb{R}^{M\times M}$ is given by $\boldsymbol{\Psi}_2 = \sum_{i=1}^{N-L} \boldsymbol{\Psi}_2^i$, where each element of the matrices $\boldsymbol{\Psi}_2^i \in \mathbb{R}^{M\times M}$ are given by

$$\begin{aligned}\left[\Psi_2^i\right]_{jj'} &= \int_{\hat{\boldsymbol{x}}_i} k\left(\hat{\boldsymbol{x}}_i, \boldsymbol{\zeta}_j\right) k\left(\hat{\boldsymbol{x}}_i, \boldsymbol{\zeta}_{j'}\right) \mathcal{N}\left(\hat{\boldsymbol{x}}_i | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) \\ &= \sigma_f^4 \prod_{d=1}^{D} \frac{\exp\left(-\frac{w_d(\zeta_{jd}-\zeta_{j'd})^2}{4} - \frac{w_d(x_{id}-\tilde{\zeta}_d)^2}{2w_d[\boldsymbol{\Sigma}_i]_{dd}+1}\right)}{\left(2w_d[\boldsymbol{\Sigma}_i]_{dd}+1\right)^{\frac{1}{2}}}, \end{aligned} \tag{A.7}$$

where $\tilde{\zeta}_d = \frac{(\zeta_{jd}+\zeta_{j'd})}{2}$.

In the case of the robust REVARB-t method, we have slightly different statistics exclusively in the output layer $H+1$, which are calculated, also omitting the layer index,

as follows:

$$\begin{aligned}\Psi_0' &= \mathrm{Tr}\left(\boldsymbol{R}\left\langle \boldsymbol{K}_f \right\rangle_{q(\boldsymbol{x})}\right) \\ &= (N-L)\sigma_f^2 \sum_{i=1}^{N-L} \frac{a_i}{b_i}, \end{aligned} \tag{A.8}$$

$$\begin{aligned}\boldsymbol{\Psi}_1' &= \boldsymbol{R}\left\langle \boldsymbol{K}_{fz} \right\rangle_{q(\boldsymbol{x})}, \\ \text{where } \left[\boldsymbol{\Psi}_1'\right]_{ij} &= \frac{a_i}{b_i}\sigma_f^2 \prod_{d=1}^{D} \frac{\exp\left(-\frac{1}{2}\frac{w_d(x_{id}-\zeta_{jd})^2}{w_d[\boldsymbol{\Sigma}_i]_{dd}+1}\right)}{(w_d[\boldsymbol{\Sigma}_i]_{dd}+1)^{\frac{1}{2}}}, \end{aligned} \tag{A.9}$$

$$\begin{aligned}\boldsymbol{\Psi}_2' &= \left\langle \left(\boldsymbol{K}_{fz}\right)^\top \boldsymbol{R}\boldsymbol{K}_{fz} \right\rangle_{q(\boldsymbol{x})} = \sum_{i=1}^{N-L} \left[\Psi_2'^i\right]_{jj'}, \\ \text{where } \left[\Psi_2'^i\right]_{jj'} &= \frac{a_i}{b_i}\sigma_f^4 \prod_{d=1}^{D} \frac{\exp\left(-\frac{w_d(\zeta_{jd}-\zeta_{j'd})^2}{4} - \frac{w_d(x_{id}-\tilde{\zeta}_d)^2}{2w_d[\boldsymbol{\Sigma}_i]_{dd}+1}\right)}{(2w_d[\boldsymbol{\Sigma}_i]_{dd}+1)^{\frac{1}{2}}}, \end{aligned} \tag{A.10}$$

where again $\tilde{\zeta}_d = \frac{(\zeta_{jd}+\zeta_{j'd})}{2}$ and $\boldsymbol{R} = \mathrm{diag}\left(\frac{a_{L+1}}{b_{L+1}}, \cdots, \frac{a_N}{b_N}\right)$ is a diagonal matrix formed by the variational parameters a_i and b_i that come from the variational gamma distribution, as explained in Section 4.3.

A.2 Derivation of the REVARB Lower Bound

In this section we complement and detail the REVARB lower bound presentation made in Section 3.3.

A.2.1 Output Layer

We begin by tackling the first line of Eq. (3.24), related to the output layer:

$$\begin{aligned}\mathscr{L}_1 &= \sum_{i=L+1}^{N} \int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H)}\right) \log p\left(y_i \middle| f_i^{(H+1)}\right) \\ &= \sum_{i=L+1}^{N} \int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H)}\right) \\ &\left(-\frac{1}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\left(y_i^2 - 2y_i f_i^{(H+1)} + \left(f_i^{(H+1)}\right)^2\right)\right) \\ &= -\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 + \sum_{i=L+1}^{N} \int_{\boldsymbol{x},\boldsymbol{z}} q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) \\ &\left(-\frac{1}{2\sigma_{H+1}^2}\left(y_i^2 - 2y_i\left[\boldsymbol{a}_f^{(H+1)}\right]_i + \left[\boldsymbol{\Sigma}_f^{(H+1)}\right]_{ii} + \left[\boldsymbol{a}_f^{(H+1)}\right]_i^2\right)\right). \end{aligned}$$

Changing to matrix notation and integrating over $\boldsymbol{x}^H$:

$$\begin{aligned}
\mathscr{L}_1 &= -\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 + \int_{\boldsymbol{x},\boldsymbol{z}} q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) \\
&\left(-\frac{1}{2\sigma_{H+1}^2}\left(\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{y}^\top\boldsymbol{a}_f^{(H+1)} + \left(\boldsymbol{a}_f^{(H+1)}\right)^\top \boldsymbol{a}_f^{(H+1)} + \mathrm{Tr}\left(\boldsymbol{\Sigma}_f^{(H+1)}\right)\right)\right) \\
&= -\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 + \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right) \\
&\left(-\frac{1}{2\sigma_{H+1}^2}\left(\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{y}^\top\left\langle\boldsymbol{K}_{fz}^{(H+1)}\right\rangle_{q(\boldsymbol{x}^H)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)}\right.\right. \\
&+ \left(\boldsymbol{z}^{(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\left\langle\left(\boldsymbol{K}_{fz}^{(H+1)}\right)^\top\boldsymbol{K}_{fz}^{(H+1)}\right\rangle_{q(\boldsymbol{x}^H)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)} \\
&\left.\left.+ \mathrm{Tr}\left(\left\langle\boldsymbol{K}_f^{(H+1)}\right\rangle_{q(\boldsymbol{x}^H)}\right) - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\left\langle\left(\boldsymbol{K}_{fz}^{(H+1)}\right)^\top\boldsymbol{K}_{fz}^{(H+1)}\right\rangle_{q(\boldsymbol{x}^H)}\right)\right)\right).
\end{aligned}$$

Now we can rewrite $\mathscr{L}_1$:

$$\begin{aligned}
\mathscr{L}_1 &= -\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\Psi_0^{(H+1)} - \frac{1}{2\sigma_{H+1}^2}\boldsymbol{y}^\top\boldsymbol{y} \\
&+ \frac{1}{2\sigma_{H+1}^2}\mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right) + \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\mathscr{P}_1,
\end{aligned} \tag{A.11}$$

where

$$\begin{aligned}
\mathscr{P}_1 &= -\frac{1}{2\sigma_{H+1}^2}\left(-2\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)}\right. \\
&\left.+ \left(\boldsymbol{z}^{(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)}\right).
\end{aligned}$$

Instead of directly integrating out $\boldsymbol{z}^{(H+1)}$, we go back to Eq. (3.24) and collect the remaining terms containing $\boldsymbol{z}^{(H+1)}$:

$$\begin{aligned}
\mathscr{L}_1^* &= -\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\Psi_0^{(H+1)} - \frac{1}{2\sigma_{H+1}^2}\boldsymbol{y}^\top\boldsymbol{y} \\
&+ \frac{1}{2\sigma_{H+1}^2}\mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right) + \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\mathscr{P}_1 \\
&- \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\log q\left(\boldsymbol{z}^{(H+1)}\right) + \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\log p\left(\boldsymbol{z}^{(H+1)}\right) \\
&= -\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\Psi_0^{(H+1)} - \frac{1}{2\sigma_{H+1}^2}\boldsymbol{y}^\top\boldsymbol{y} \\
&+ \frac{1}{2\sigma_{H+1}^2}\mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right) - \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\log\frac{q\left(\boldsymbol{z}^{(H+1)}\right)}{\exp(\mathscr{P}_1)p\left(\boldsymbol{z}^{(H+1)}\right)},
\end{aligned} \tag{A.12}$$

where the last term is a "KL-like" term. The optimal distribution $q^*\left(\boldsymbol{z}^{(H+1)}\right)$ that maximizes the bound is proportional to the denominator inside the logarithm:

$$q^*\left(\boldsymbol{z}^{(H+1)}\right) \propto \exp(\mathscr{P}_1) p\left(\boldsymbol{z}^{(H+1)}\right) \tag{A.13}$$

$$\begin{aligned}\log q^*\left(\boldsymbol{z}^{(H+1)}\right) &\propto \mathscr{P}_1 + \log p\left(\boldsymbol{z}^{(H+1)}\right) \\ &\propto -\frac{1}{2\sigma_{H+1}^2}\left(-2\boldsymbol{y}^\top \boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)}\right. \\ &+ \left(\boldsymbol{z}^{(H+1)}\right)^\top \left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)}\Big) \\ &- \frac{1}{2}\left(\boldsymbol{z}^{(H+1)}\right)^\top \left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)},\end{aligned} \tag{A.14}$$

where we have omitted some terms that did not contain $\boldsymbol{z}^{(H+1)}$. Rearranging the expression we get:

$$\begin{aligned}\log q^*\left(\boldsymbol{z}^{(H+1)}\right) &\propto -\frac{1}{2\sigma_{H+1}^2} 2\boldsymbol{y}^\top \boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)} \\ &- \frac{1}{2}\left(\boldsymbol{z}^{(H+1)}\right)^\top \left(\frac{1}{\sigma_{H+1}^2}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1} + \left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\right)\boldsymbol{z}^{(H+1)},\end{aligned}$$

which is proportional to the logarithm of a Gaussian distribution. The moments of such distribution can be found by "completing the square" (BISHOP, 2006):

$$\begin{aligned}q^*\left(\boldsymbol{z}^{(H+1)}\right) &= \mathscr{N}\left(\boldsymbol{z}^{(H+1)} \middle| \boldsymbol{m}^{(H+1)}, \boldsymbol{S}^{(H+1)}\right), \\ \boldsymbol{S}^{(H+1)} &= \left(\frac{1}{\sigma_{H+1}^2}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1} + \left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\right)^{-1}, \\ \boldsymbol{m}^{(H+1)} &= \frac{1}{\sigma_{H+1}^2}\boldsymbol{S}^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top \boldsymbol{y}.\end{aligned}$$

Rewriting these expressions we get the optimized moments of $q^*\left(\boldsymbol{z}^{(H+1)}\right)$:

$$\boldsymbol{S}^{(H+1)} = \boldsymbol{K}_z^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\boldsymbol{K}_z^{(H+1)}, \tag{A.15}$$

$$\boldsymbol{m}^{(H+1)} = \frac{1}{\sigma_{H+1}^2}\boldsymbol{K}_z^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top \boldsymbol{y}. \tag{A.16}$$

Replacing the optimal distribution $q^*\left(\boldsymbol{z}^{(H+1)}\right)$ back in Eq. (A.14) we get:

$$\begin{aligned}\log q^*\left(\boldsymbol{z}^{(H+1)}\right) &\propto \mathscr{P}_1 + \log p\left(\boldsymbol{z}^{(H+1)}\right)\\ &= \log \mathscr{N}\left(\boldsymbol{z}^{(H+1)}\middle|\boldsymbol{m}^{(H+1)}\boldsymbol{S}^{(H+1)}\right)\\ &\underbrace{-\frac{1}{2}\log\left|\boldsymbol{K}_z^{H+1}\right| + \frac{1}{2}\log\left|\boldsymbol{S}^{(H+1)}\right| + \frac{1}{2}\left(\boldsymbol{m}^{(H+1)}\right)^\top\left(\boldsymbol{S}^{(H+1)}\right)^{-1}\boldsymbol{m}^{(H+1)}}_{\mathscr{P}_2}.\end{aligned}$$

The terms of $\mathscr{P}_2$ can be found using the optimized moments of $q^*\left(\boldsymbol{z}^{(H+1)}\right)$:

$$\begin{aligned}\frac{1}{2}\log\left|\boldsymbol{S}^{(H+1)}\right| &= \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\boldsymbol{K}_z^{(H+1)}\right|\\ &= \log\left|\boldsymbol{K}_z^{(H+1)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right|,\end{aligned}$$

$$\begin{aligned}&\frac{1}{2}\left(\boldsymbol{m}^{(H+1)}\right)^\top\left(\boldsymbol{S}^{(H+1)}\right)^{-1}\boldsymbol{m}^{(H+1)} =\\ &\frac{1}{2\left(\sigma_{H+1}^2\right)^2}\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top\boldsymbol{y}.\end{aligned}$$

Now we can rewrite the last term of Eq. (A.12):

$$\begin{aligned}-\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\log\frac{q\left(\boldsymbol{z}^{(H+1)}\right)}{\exp(\mathscr{P}_1)p\left(\boldsymbol{z}^{(H+1)}\right)} &\leq \int_{\boldsymbol{z}} q^*\left(\boldsymbol{z}^{(H+1)}\right)\log\frac{\exp(\mathscr{P}_1)p\left(\boldsymbol{z}^{(H+1)}\right)}{q^*\left(\boldsymbol{z}^{(H+1)}\right)}\\ &\leq \log\int_{\boldsymbol{z}}\exp(\mathscr{P}_1)p\left(\boldsymbol{z}^{(H+1)}\right)\\ &\leq \log\int_{\boldsymbol{z}}\mathscr{N}\left(\boldsymbol{z}^{(H+1)}\middle|\boldsymbol{m}^{(H+1)}\boldsymbol{S}^{(H+1)}\right)\exp(\mathscr{P}_2)\\ &\leq \mathscr{P}_2,\end{aligned}$$

where we have reversed the Jensen's inequality and made the bound tighter. This trick is also explained by King and Lawrence (2006).

Finally, Eq. (A.12) becomes:

$$\begin{aligned}\mathscr{L}_1^* = &-\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\left(\boldsymbol{y}^\top\boldsymbol{y} + \Psi_0^{(H+1)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right)\right)\\ &+\frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right|\\ &+\frac{1}{2\left(\sigma_{H+1}^2\right)^2}\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top\boldsymbol{y}.\end{aligned} \tag{A.17}$$

A.2.2 Hidden Layers

We now tackle the second line of Eq. (3.24), related to the hidden layers. Note that its format resembles the first line we have just solved, so we follow similar procedures of the previous section:

$$
\begin{aligned}
\mathscr{L}_2 &= \sum_{i=L+1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}}\left(\prod_{h'=1}^{H} q\left(\boldsymbol{x}^{(h')}\right)\right) q\left(\boldsymbol{z}^{(h)}\right) p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\log p\left(x_i^{(h)}\middle|f_i^{(h)}\right)\\
&= -\frac{N-L}{2}\sum_{h=1}^{H}\log 2\pi\sigma_h^2 + \sum_{h=1}^{H}\int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}}\left(\prod_{h'=1}^{H} q\left(\boldsymbol{x}^{(h')}\right)\right) q\left(\boldsymbol{z}^{(h)}\right) p\left(\boldsymbol{f}^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\\
&\left(-\frac{1}{2\sigma_h^2}\left(\left(\boldsymbol{x}^{(h)}\right)^\top\boldsymbol{x}^{(h)} - 2\left(\boldsymbol{x}^{(h)}\right)^\top\boldsymbol{f}^{(h)} + \left(\boldsymbol{f}^{(h)}\right)^\top\boldsymbol{f}^{(h)}\right)\right)\\
&= -\frac{N-L}{2}\sum_{h=1}^{H}\log 2\pi\sigma_h^2 + \sum_{h=1}^{H}\int_{\boldsymbol{x},\boldsymbol{z}}\left(\prod_{h'=1}^{H} q\left(\boldsymbol{x}^{(h')}\right)\right) q\left(\boldsymbol{z}^{(h)}\right)\\
&\left(-\frac{1}{2\sigma_h^2}\left(\left(\boldsymbol{x}^{(h)}\right)^\top\boldsymbol{x}^{(h)} - 2\left(\boldsymbol{x}^{(h)}\right)^\top\boldsymbol{a}_f^{(h)} + \left(\boldsymbol{a}_f^{(h)}\right)^\top\boldsymbol{a}_f^{(h)} + \mathrm{Tr}\left(\boldsymbol{\Sigma}_f^{(h)}\right)\right)\right)\\
&= -\frac{N-L}{2}\sum_{h=1}^{H}\log 2\pi\sigma_h^2 + \sum_{h=1}^{H}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\\
&\left(-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} - 2\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{z}^{(h)}\right.\right.\\
&\left.\left. + \left(\boldsymbol{z}^{(h)}\right)^\top\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{z}^{(h)} + \Psi_0^{(h)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right)\\
&= -\frac{N-L}{2}\sum_{h=1}^{H}\log 2\pi\sigma_h^2\\
&+ \sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} + \Psi_0^{(h)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right.\\
&\left. + \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\mathscr{T}_1^{(h)}\right\},
\end{aligned}
$$

where

$$
\begin{aligned}
\mathscr{T}_1^{(h)} = -\frac{1}{2\sigma_h^2}&\left(-2\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{z}^{(h)}\right.\\
&\left. + \left(\boldsymbol{z}^{(h)}\right)^\top\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{z}^{(h)}\right).
\end{aligned}
$$

Note that the integration of the latent variables $\boldsymbol{x}^{(h)}$ followed the statistics defined in Eq. (3.27) and detailed in the Appendix A.1.

As in the previous section, we can get the optimal distribution $q^*\left(\boldsymbol{z}^{(h)}\right)$:

$$q^*\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{m}^{(h)}, \boldsymbol{S}^{(h)}\right), \tag{A.18}$$

$$\boldsymbol{S}^{(h)} = \boldsymbol{K}_z^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\boldsymbol{K}_z^{(h)}, \tag{A.19}$$

$$\boldsymbol{m}^{(h)} = \frac{1}{\sigma_h^2}\boldsymbol{K}_z^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top \boldsymbol{\mu}^{(h)}. \tag{A.20}$$

Note that the difference to the output layer is that we replaced $\boldsymbol{y}$ by $\boldsymbol{\mu}^{(h)}$.

Following the same argument we used for the output layer, we get:

$$\begin{aligned}
\mathscr{L}_2^* = &-\frac{N-L}{2}\sum_{h=1}^{H}\log 2\pi\sigma_h^2 \\
&+\sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} + \Psi_0^{(h)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right. \\
&+\frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right| \\
&\left.+\frac{1}{2\left(\sigma_{H+1}^2\right)^2}\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top\boldsymbol{\mu}^{(h)}\right\}.
\end{aligned} \tag{A.21}$$

A.2.3 Final REVARB Lower Bound

Putting together Eqs. (A.17) and (A.21) and adding the remaining terms from Eq. (3.24) we get the final form of the REVARB lower bound:

$$\begin{aligned}
\log p(\boldsymbol{y}) \geq &-\frac{N-L}{2}\sum_{h=1}^{H+1}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_{H+1}^2}\left(\boldsymbol{y}^\top\boldsymbol{y} + \Psi_0^{(H+1)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right)\right) \\
&+\frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right| \\
&+\frac{1}{2(\sigma_{H+1}^2)^2}\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top\boldsymbol{y} \\
&+\sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} + \Psi_0^{(h)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right. \\
&+\frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right| \\
&+\frac{1}{2(\sigma_h^2)^2}\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} \\
&\left.-\sum_{i=1}^{N}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right)\right\}.
\end{aligned}$$

The terms in the last line of the final bound involves only Gaussian distributions and are, hence, tractable. They are calculated as follows:

$$\int_{x_i^{(h)}} q\left(x_i^{(h)}\right) \log q\left(x_i^{(h)}\right) = -\frac{1}{2}\log 2\pi\lambda_i^{(h)} - \frac{1}{2},$$
$$\int_{x_i^{(h)}} q\left(x_i^{(h)}\right) \log p\left(x_i^{(h)}\right) = -\frac{1}{2}\log 2\pi\lambda_{0i}^{(h)} - \frac{1}{2\lambda_{0i}^{(h)}}\left[\lambda_i^{(h)} + \left(\mu_i^{(h)}\right)^2 - 2\mu_i^{(h)}\mu_{0i}^{(h)} + \left(\mu_{0i}^{(h)}\right)^2\right].$$

A.3 Derivation of the GP-RLARX Variational Lower Bound

The lower bound to the marginal log-likelihood $\log p(\boldsymbol{y})$ of the GP-RLARX model in Eq. (4.20) can be written as follows:

$$\begin{aligned}\log p(\boldsymbol{y}) \geq & \sum_{i=L+1}^{N} \int_{\boldsymbol{\tau},\boldsymbol{x}} q(\boldsymbol{\tau})q(\boldsymbol{x}) \log p(y_i|x_i,\tau_i) - \mathrm{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau})) \\ & + \sum_{i=L+1}^{N} \int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q(\boldsymbol{x})q(\boldsymbol{z})p(f_i|\boldsymbol{z},\hat{\boldsymbol{x}}_i) \log p(x_i|f_i) \\ & - \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log q(\boldsymbol{z}) + \int_{\boldsymbol{z}} q(\boldsymbol{z}) \log p(\boldsymbol{z}) \\ & - \sum_{i=1}^{N} \int_{\boldsymbol{x}} q(x_i) \log q(x_i) + \sum_{i=1}^{L} \int_{\boldsymbol{x}} q(x_i) \log p(x_i),\end{aligned} \tag{A.22}$$

where the KL-divergence $\mathrm{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau}))$ between two factorized gamma distributions was defined in Eq. (4.33).

We start to tackle Eq. (A.22) from the integral that contains the observations $\boldsymbol{y}$, where we first variationally integrate $\boldsymbol{\tau}$ and then $\boldsymbol{x}$:

$$\begin{aligned}\mathscr{L}_1 &= \sum_{i=L+1}^{N} \int_{\boldsymbol{\tau},\boldsymbol{x}} q(\boldsymbol{\tau})q(\boldsymbol{x}) \log p(y_i|x_i,\tau_i) \\ &= \sum_{i=L+1}^{N} \int_{\boldsymbol{\tau},\boldsymbol{x}} q(\boldsymbol{\tau})q(\boldsymbol{x}) \left[-\frac{1}{2}\log 2\pi + \frac{1}{2}\log\tau_i - \frac{\tau_i}{2}\left(y_i^2 - 2y_i x_i + x_i^2\right)\right] \\ &= -\frac{N-L}{2}\log 2\pi + \frac{1}{2}\sum_{i=L+1}^{N}(\psi(a_i) - \log b_i) + \sum_{i=L+1}^{N} \int_{\boldsymbol{x}} q(\boldsymbol{x}) \left[-\frac{a_i}{2b_i}\left(y_i^2 - 2y_i x_i + x_i^2\right)\right] \\ &= -\frac{N-L}{2}\log 2\pi + \frac{1}{2}\sum_{i=L+1}^{N}(\psi(a_i) - \log b_i) + -\frac{1}{2}\sum_{i=L+1}^{N}\left[\frac{a_i}{b_i}\left(y_i^2 - 2y_i\mu_i + \lambda_i + \mu_i^2\right)\right],\end{aligned} \tag{A.23}$$

where we have applied the following properties of the gamma distribution:

$$\tau_i \sim \Gamma(a_i, b_i), \quad \mathbb{E}\{\tau_i\} = \frac{a_i}{b_i}, \quad \mathbb{E}\{\log\tau_i\} = \psi(a_i) - \log b_i.$$

The second line in Eq. (A.23) actually presents similar integrals to the hidden layers of the REVARB method presented in Appendix A.2.2 if we simply ignore the layer indexes. Consequently, the GP-RLARX version of Eq. (A.21) will be given by

$$\begin{aligned}\mathscr{L}_2^* = &-\frac{N-L}{2}\log 2\pi\sigma_x^2 - \frac{1}{2\sigma_x^2}\left[\sum_{i=L+1}^{N}\left(\lambda_i+\mu_i^2\right)+\Psi_0-\operatorname{Tr}\left(\boldsymbol{K}_z^{-1}\boldsymbol{\Psi}_2\right)\right] \\ &+\frac{1}{2}\log|\boldsymbol{K}_z| - \frac{1}{2}\log\left|\boldsymbol{K}_z+\frac{1}{\sigma_x^2}\boldsymbol{\Psi}_2\right| + \frac{1}{2\left(\sigma_x^2\right)^2}\left(\boldsymbol{\mu}\right)^\top\boldsymbol{\Psi}_1\left(\boldsymbol{K}_z+\frac{1}{\sigma_x^2}\boldsymbol{\Psi}_2\right)^{-1}\boldsymbol{\Psi}_1^\top\boldsymbol{\mu}.\end{aligned} \tag{A.24}$$

The final lower bound for the GP-RLARX model is finally given by summing Eqs. (A.23) and (A.24) and including the remaining terms of Eq. (A.22):

$$\begin{aligned}\log p(\boldsymbol{y}) \geq &-\frac{N-L}{2}\log 2\pi\sigma_x^2 - \frac{N-L}{2}\log 2\pi + \frac{1}{2}\sum_{i=L+1}^{N}\left(\psi(a_i)-\log b_i\right) \\ &-\frac{1}{2}\sum_{i=L+1}^{N}\left[\frac{a_i}{b_i}\left(y_i^2-2y_i\mu_i+\lambda_i+\mu_i^2\right)\right] - \frac{1}{2\sigma_x^2}\left[\sum_{i=1}^{N}\left(\lambda_i+\mu_i^2\right)+\Psi_0-\operatorname{Tr}(\boldsymbol{K}_z^{-1}\boldsymbol{\Psi}_2)\right] \\ &+\frac{1}{2}\log|\boldsymbol{K}_z| - \frac{1}{2}\log\left|\boldsymbol{K}_z+\frac{1}{\sigma_x^2}\boldsymbol{\Psi}_2\right| + \frac{1}{2(\sigma_x^2)^2}\boldsymbol{\mu}^\top\boldsymbol{\Psi}_1\left(\boldsymbol{K}_z+\frac{1}{\sigma_x^2}\boldsymbol{\Psi}_2\right)^{-1}\boldsymbol{\Psi}_1^\top\boldsymbol{\mu} \\ &-\sum_{i=1}^{N}\int_{x_i} q(x_i)\log q(x_i) + \sum_{i=1}^{L}\int_{x_i} q(x_i)\log p(x_i) - \operatorname{KL}(q(\boldsymbol{\tau})\,\|\,p(\boldsymbol{\tau})).\end{aligned} \tag{A.25}$$

A.4 Derivation of the REVARB-t Lower Bound

The REVARB-t lower bound to the marginal log-likelihood $\log p(\boldsymbol{y})$ in Eq. (4.42) can be detailed as follows:

$$\begin{aligned}\log p(\boldsymbol{y}) \geq &\sum_{i=L+1}^{N}\left\{\int_{\boldsymbol{\tau},\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q(\boldsymbol{\tau})q\left(\boldsymbol{x}^{(H)}\right)q\left(\boldsymbol{z}^{(H+1)}\right)\right. \\ &\left.p\left(f_i^{(H+1)}\middle|\boldsymbol{z}^{(H+1)},\hat{\boldsymbol{x}}_i^{(H)}\right)\log p\left(y_i\middle|f_i^{(H+1)},\tau_i\right)\right\} - \operatorname{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau})) \\ &+\sum_{i=L+1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}}\left(\prod_{h'=1}^{H}q\left(\boldsymbol{x}^{(h')}\right)\right)q\left(\boldsymbol{z}^{(h)}\right)p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\log p\left(x_i^{(h)}\middle|f_i^{(h)}\right) \\ &-\sum_{h=1}^{H+1}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log q\left(\boldsymbol{z}^{(h)}\right) + \sum_{h=1}^{H+1}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log p\left(\boldsymbol{z}^{(h)}\right) \\ &-\sum_{i=1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right).\end{aligned} \tag{A.26}$$

The bound in Eq. (A.26) differs from the original REVARB bound only on the terms related to the output layer and the included KL-divergence $\operatorname{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau}))$ defined in Eq.

(4.33). It also differs from the GP-RLAX bound in Eq. (A.22), since it includes multiple transition (hidden) layers and a GP prior in the observation layer.

We proceed by variationally integrating out $\boldsymbol{\tau}$ from the output layer, similar to the GP-RLARX model in the previous section:

$$\begin{aligned}
\mathscr{L}_1 = & \sum_{i=L+1}^{N} \left\{ \int_{\boldsymbol{\tau},\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q(\boldsymbol{\tau}) q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) \right. \\
& \left. p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H)}\right) \log p\left(y_i \middle| f_i^{(H+1)}, \tau_i\right) \right\} \\
= & \sum_{i=L+1}^{N} \left\{ \int_{\boldsymbol{\tau},\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q(\boldsymbol{\tau}) q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H)}\right) \right. \\
& \left. \left(-\frac{1}{2}\log 2\pi + \frac{1}{2}\log \tau_i - \frac{\tau_i}{2}\left(y_i^2 - 2y_i f_i^{(H+1)} + \left(f_i^{(H+1)}\right)^2 \right) \right) \right\} \\
= & -\frac{N-L}{2}\log 2\pi + \frac{1}{2}\sum_{i=L+1}^{N} \left(\psi(a_i) - \log b_i\right) \\
& + \sum_{i=L+1}^{N} \left\{ \int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H)}\right) \right. \\
& \left. \left(-\frac{a_i}{2b_i}\left(y_i^2 - 2y_i f_i^{(H+1)} + \left(f_i^{(H+1)}\right)^2 \right) \right) \right\}.
\end{aligned}$$

If we define $\boldsymbol{R} = \text{diag}\left(\frac{a_{L+1}}{b_{L+1}}, \cdots, \frac{a_N}{b_N}\right)$ and follow the same steps we did for REVARB in the Appendix A.2 in order to integrate $\boldsymbol{f}^{(H+1)}$ and $\boldsymbol{x}^{(H)}$, we get the REVARB-t version of Eq. (A.17):

$$\begin{aligned}
\mathscr{L}_1^* = & -\frac{N-L}{2}\log 2\pi + \frac{1}{2}\sum_{i=L+1}^{N} \left(\psi(a_i) - \log b_i\right) \\
& - \frac{1}{2}\left(\boldsymbol{y}^\top \boldsymbol{R}\boldsymbol{y} + \Psi_0'^{(H+1)} - \text{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1} \boldsymbol{\Psi}_2'^{(H+1)} \right) \right) \\
& + \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)} + \boldsymbol{\Psi}_2'^{(H+1)}\right| \\
& + \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{\Psi}_1'^{(H+1)} \left(\boldsymbol{K}_z^{(H+1)} + \boldsymbol{\Psi}_2'^{(H+1)}\right)^{-1} \left(\boldsymbol{\Psi}_1'^{(H+1)}\right)^\top \boldsymbol{y},
\end{aligned} \tag{A.27}$$

where we have used the modified statistics defined in the Eqs. (4.49), (4.50) and (4.51) and detailed in the Appendix A.1.

The bound terms related to the hidden layers are equal to the ones for the REVARB with Gaussian likelihood, detailed in the Appendix A.2.2. Thus, the final lower

bound for REVARB-t is given by

$$\begin{aligned}
\log p(\boldsymbol{y}) \geq & -\frac{N-L}{2}\sum_{h=1}^{H}\log 2\pi\sigma_h^2 - \frac{N-L}{2}\log 2\pi + \frac{1}{2}\sum_{i=L+1}^{N}\left(\psi(a_i)-\log b_i\right) \\
& -\frac{1}{2}\left(\boldsymbol{y}^\top \boldsymbol{R}\boldsymbol{y} + \Psi_0^{\prime(H+1)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{\prime(H+1)}\right)\right) \\
& +\frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)} + \boldsymbol{\Psi}_2^{\prime(H+1)}\right| \\
& +\frac{1}{2}\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{\prime(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \boldsymbol{\Psi}_2^{\prime(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{\prime(H+1)}\right)^\top\boldsymbol{y} - \mathrm{KL}(q(\boldsymbol{\tau})||p(\boldsymbol{\tau})) \\
& +\sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} + \Psi_0^{(h)} - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right. \\
& +\frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right| \\
& +\frac{1}{2(\sigma_h^2)^2}\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} \\
& \left.-\sum_{i=1}^{N}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right)\right\}.
\end{aligned} \tag{A.28}$$

A.5 Derivation of the S-REVARB Lower Bound

In this section we detail the S-REVARB lower bound presentation made in Section 5.3. Although it is derived from the same REVARB initial bound, this time we aim for a non-collapsed bound, explicitly parametrized by the moments of the distributions $q\left(\boldsymbol{z}^{(h)}\right)$ in each layer. This approach will result in a fully factorized expression and enable stochastic updates.

A.5.1 Output Layer

Before factorization, the S-REVARB lower bound is the same in Eq. (3.24). Similar to REVARB, we will first consider the output layer, expressed in Eq. (A.11) and replicated below:

$$\begin{aligned}
\mathscr{L}_1 = & -\frac{N-L}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\Psi_0^{(H+1)} - \frac{1}{2\sigma_{H+1}^2}\boldsymbol{y}^\top\boldsymbol{y} \\
& +\frac{1}{2\sigma_{H+1}^2}\mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right) \\
& -\frac{1}{2\sigma_{H+1}^2}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\left[-2\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)}\right. \\
& \left.+\left(\boldsymbol{z}^{(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)}\right].
\end{aligned}$$

Such expression can be rewritten as a sum of terms related to each observation as follows:

$$\begin{aligned}\mathscr{L}_1 = \sum_{i=L+1}^{N} \Bigg\{ & -\frac{1}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\left[\Psi_0^{i(H+1)} + y_i^2 - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\right)\right] \\ & - \frac{1}{2\sigma_{H+1}^2}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\Bigg[-2y_i\left(\boldsymbol{\Psi}_1^{i(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{z}^{(H+1)} \\ & + \mathrm{Tr}\left(\boldsymbol{z}^{(H+1)}\left(\boldsymbol{z}^{(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\right)\Bigg]\Bigg\},\end{aligned}$$

where we have redefined the statistics $\Psi_0^{i(h)}$, $\boldsymbol{\Psi}_1^{i(h)}$ and $\boldsymbol{\Psi}_2^{i(h)}$ with respect to each sample (or mini-batch), given by

$$\begin{aligned}\Psi_0^{i(h)} &= \left\langle \left[\boldsymbol{K}_f^{(h)}\right]_{ii}\right\rangle_{q(\cdot)^{(h)}}, \\ \boldsymbol{\Psi}_1^{i(h)} &= \left\langle \left[\boldsymbol{K}_{fz}^{(h)}\right]_i\right\rangle_{q(\cdot)^{(h)}}, \\ \boldsymbol{\Psi}_2^{i(h)} &= \left\langle \left[\boldsymbol{K}_{fz}^{(h)}\right]_i\left[\boldsymbol{K}_{fz}^{(h)}\right]_i^\top\right\rangle_{q(\cdot)^{(h)}}.\end{aligned}$$

The expressions are similar to the REVARB statistics presented in the Appendix A.1.

Now we can integrate out $\boldsymbol{z}^{(H+1)}$ by explicitly defining the variational distribution $q\left(\boldsymbol{z}^{(H+1)}\right) = \mathscr{N}\left(\boldsymbol{z}^{(H+1)}\middle|\boldsymbol{m}^{(H+1)}, \boldsymbol{S}^{(H+1)}\right)$, where $\boldsymbol{m}^{(H+1)} \in \mathbb{R}^M$ and $\boldsymbol{S}^{(H+1)} \in \mathbb{R}^{M\times M}$:

$$\begin{aligned}\mathscr{L}_1 = \sum_{i=L+1}^{N} \Bigg\{ & -\frac{1}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\left[\Psi_0^{i(H+1)} + y_i^2 - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\right)\right] \\ & + \frac{1}{\sigma_{H+1}^2}y_i\left(\boldsymbol{\Psi}_1^{i(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{m}^{(H+1)} \\ & - \frac{1}{2\sigma_{H+1}^2}\mathrm{Tr}\left(\left(\boldsymbol{S}^{(H+1)} + \boldsymbol{m}^{(H+1)}\left(\boldsymbol{m}^{(H+1)}\right)^\top\right)\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\right)\Bigg\}.\end{aligned} \tag{A.29}$$

We also need to include the negative KL term recognized in Eq. (3.24), which involve only Gaussians and is, hence, tractable:

$$\begin{aligned}& -\mathrm{KL}\left(q\left(\boldsymbol{z}^{(H+1)}\right)\middle\|p\left(\boldsymbol{z}^{(H+1)}\right)\right) = \\ & -\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\log q\left(\boldsymbol{z}^{(H+1)}\right) + \int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(H+1)}\right)\log p\left(\boldsymbol{z}^{(H+1)}\right) \\ & = -\frac{1}{2}\Bigg[\mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{S}^{(H+1)}\right) + \left(\boldsymbol{m}^{(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{m}^{(H+1)} \\ & - M + \log\left|\boldsymbol{K}_z^{(H+1)}\right| - \log\left|\boldsymbol{S}^{(H+1)}\right|\Bigg].\end{aligned} \tag{A.30}$$

Finally, the non-collapsed lower bound terms related to the output layer become:

$$\begin{aligned}\mathscr{L}_1^* = \sum_{i=L+1}^{N} \Bigg\{ & -\frac{1}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\left[\Psi_0^{i(H+1)} + y_i^2 - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\right)\right] \\ & + \frac{1}{\sigma_{H+1}^2} y_i \left(\boldsymbol{\Psi}_1^{i(H+1)}\right)^\top \left(\boldsymbol{K}_z^{(H+1)}\right)^{-1} \boldsymbol{m}^{(H+1)} \\ & - \frac{1}{2\sigma_{H+1}^2}\mathrm{Tr}\left(\left(\boldsymbol{S}^{(H+1)} + \boldsymbol{m}^{(H+1)}\left(\boldsymbol{m}^{(H+1)}\right)^\top\right)\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\right)\Bigg\} \\ & - \frac{1}{2}\Bigg[\mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{S}^{(H+1)}\right) + \left(\boldsymbol{m}^{(H+1)}\right)^\top\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{m}^{(H+1)} \\ & - M + \log\left|\boldsymbol{K}_z^{(H+1)}\right| - \log\left|\boldsymbol{S}^{(H+1)}\right|\Bigg].\end{aligned} \tag{A.31}$$

As expected, the new expression is fully factorized with respect to the training samples.

A.5.2 Hidden layers

The lower bound terms related to the hidden layers resemble the ones of the output layer, but the outputs of the hidden layer are latent and must be also integrated out, similar to the standard REVARB approach.

In order to obtain the new expressions, we start by replacing the terms in Eq. (A.29) which contain y_i by $\mathbb{E}\left\{x_i^{(h)}\right\} = \mu_i^{(h)}$ and y_i^2 by $\mathbb{E}\left\{\left(x_i^{(h)}\right)^2\right\} = \left(\mu_i^{(h)}\right)^2 + \lambda_i^{(h)}$. Thus, we have:

$$\begin{aligned}\mathscr{L}_2 = \sum_{i=L+1}^{N} \Bigg\{ & -\frac{1}{2}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_h^2}\left[\Psi_0^{i(h)} + \left(\mu_i^{(h)}\right)^2 + \lambda_i - \mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{i(h)}\right)\right] \\ & + \frac{1}{2\sigma_h^2}\mu_i^{(h)}\left(\boldsymbol{\Psi}_1^{i(h)}\right)^\top\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)} \\ & - \frac{1}{2\sigma_h^2}\mathrm{Tr}\left(\left(\boldsymbol{S}^{(h)} + \boldsymbol{m}^{(h)}\left(\boldsymbol{m}^{(h)}\right)^\top\right)\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{i(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\right)\Bigg\}.\end{aligned}$$

Besides the KL term, which is identical as before in Eq. (A.30), we also need to include the entropy and the terms associated to the priors in Eq. (3.24) to obtain the bound expression related to the hidden layers:

$$\begin{aligned}\mathscr{L}_2^* = \mathscr{L}_2 & - \frac{1}{2}\Bigg(\mathrm{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{S}^{(h)}\right) + \left(\boldsymbol{m}^{(h)}\right)^\top\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)} \\ & - M + \log\left|\boldsymbol{K}_z^{(h)}\right| - \log\left|\boldsymbol{S}^{(h)}\right|\Bigg) \\ & - \sum_{i=1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right).\end{aligned} \tag{A.32}$$

A.5.3 Final S-REVARB Lower Bound

The final S-REVARB non-collapsed lower bound is obtained by putting together Eqs. (A.31) and (A.32):

$$
\begin{aligned}
\log p(\boldsymbol{y}) \geq \sum_{i=L+1}^{N} \Bigg\{ & -\frac{1}{2}\log 2\pi\sigma_{H+1}^2 - \frac{1}{2\sigma_{H+1}^2}\left(\Psi_0^{i(H+1)} + y_i^2 - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\right)\right) \\
& + \frac{1}{\sigma_{H+1}^2} y_i \left(\boldsymbol{\Psi}_1^{i(H+1)}\right)^{\top}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{m}^{(H+1)} \\
& - \frac{1}{2\sigma_{H+1}^2}\operatorname{Tr}\left(\left(\boldsymbol{S}^{(H+1)} + \boldsymbol{m}^{(H+1)}\left(\boldsymbol{m}^{(H+1)}\right)^{\top}\right)\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{i(H+1)}\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\right) \\
& + \sum_{h=1}^{H}\Bigg[-\frac{1}{2}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_h^2}\left(\Psi_0^{i(h)} + \left(\mu_i^{(h)}\right)^2 + \lambda_i - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{i(h)}\right)\right) \\
& + \frac{1}{\sigma_h^2}\mu_i^{(h)}\left(\boldsymbol{\Psi}_1^{i(h)}\right)^{\top}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)} \\
& - \frac{1}{2\sigma_h^2}\operatorname{Tr}\left(\left(\boldsymbol{S}^{(h)} + \boldsymbol{m}^{(h)}\left(\boldsymbol{m}^{(h)}\right)^{\top}\right)\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{i(h)}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\right) \\
& - \int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right)\Bigg]\Bigg\} \\
& - \frac{1}{2}\sum_{h=1}^{H+1}\Bigg[\operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{S}^{(h)}\right) + \left(\boldsymbol{m}^{(h)}\right)^{\top}\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{m}^{(h)} - M \\
& + \log\left|\boldsymbol{K}_z^{(h)}\right| - \log\left|\boldsymbol{S}^{(h)}\right|\Bigg] \\
& + \sum_{i=1}^{L}\sum_{h=1}^{H}\left[\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right) - \int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right)\right].
\end{aligned}
\tag{A.33}
$$